



TESIS - EE185401

***TRACKING TRAJECTORY AUTOMATED GUIDED
VEHICLE MENGGUNAKAN FUZZY KONTROLER
PADA RUTE MAGNETIC SPOT***

MAMAT SEPTYAN
07111650020002

DOSEN PEMBIMBING
Dr. Trihastuti Agustinah, ST., MT

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019



TESIS - EE185401

***TRACKING TRAJECTORY AUTOMATED GUIDED
VEHICLE MENGGUNAKAN FUZZY KONTROLER
PADA RUTE *MAGNETIC SPOT****

MAMAT SEPTYAN
07111650020002

DOSEN PEMBIMBING
Dr. Trihastuti Agustinah, ST., MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

di
Institut Teknologi Sepuluh Nopember

oleh:

Mamat Septyan
07111650020002

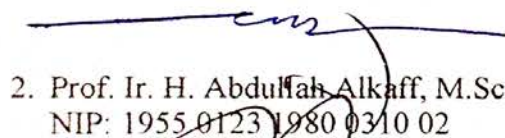
Tanggal Ujian : 28 Desember 2018
Periode Wisuda: Maret 2019

Disetujui oleh:



1. Dr. Trihastuti Agustinah, ST., MT
NIP: 1968 0812 1994 0320 01

(Pembimbing I)



2. Prof. Ir. H. Abdulfah Alkaff, M.Sc., Ph.D (Penguji)
NIP: 1955 0123 1980 0310 02

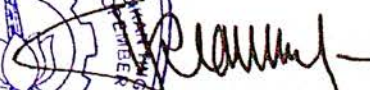


3. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
NIP: 1959 0219 1986 1010 01

(Penguji)



Dekan Fakultas Teknologi Elektro



Tri Anief Sardjono, S.T., M.T.
NIP: 1970 0212 1995 1210 01

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "*TRACKING TRAJECTORY AUTOMATED GUIDED VEHICLE MENGGUNAKAN FUZZY KONTROLER PADA RUTE MAGNETIC SPOT*" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Maret 2018



Mamat Septyan

NRP. 07111650020002

Halaman ini sengaja dikosongkan

TRACKING TRAJECTORY AUTOMATED GUIDED VEHICLE MENGUNAKAN FUZZY KONTROLER PADA RUTE MAGNETIC SPOT

Nama mahasiswa : Mamat Septyan
NRP : 07111650020002
Pembimbing : 1. Dr. Trihastuti Agustinah, ST., MT.

ABSTRAK

Automated Guided Vehicle (AGV) merupakan moda transportasi yang penting dalam industri. Untuk mencapai fleksibilitas dalam menjalankan tugas yang berulang-ulang, dibutuhkan sistem kontrol yang memadai dalam pengaturan transportasi sehingga teknologi ini banyak digunakan sebagai objek penelitian. Salah satu area penelitian yang masih terbuka adalah permasalahan kontrol *tracking* terutama untuk *trajectory* referensi dengan perubahan sudut yang besar. Terdapatnya batasan pergerakan untuk AGV dengan *nonholonomic constraint* mengakibatkan *tracking* pada *nonsmooth trajectory* tidak mudah untuk dilakukan.

Penelitian ini membahas pengembangan kontroler fuzzy Takagi-Sugeno dan *dynamic nonlinear feedback* untuk meminimumkan *error tracking* pada *nonsmooth trajectory*. Kontroler dirancang berdasarkan fuzzy Takagi-Sugeno model dan Parallel Distributed Compensation (PDC) dengan *gain* dan matriks stabilitas dihitung menggunakan Linear Matrix Inequality (LMI). *Feedback* didapatkan dari dua pengukuran sebagai *input* utama kontroler dan sebagai faktor koreksi. Perhitungan torsi model dinamika AGV diperoleh dari *input auxiliary velocity* dan *dynamic nonlinear feedback*.

Hasil simulasi menunjukkan kontroler dengan satu pengukuran menghasilkan *steady state error tracking* dengan nilai *settling time* kurang dari 15 s pada semua *trajectory* referensi kecuali untuk *error* orientasi dimana nilai *settling time* sebesar 31 s karena di setiap titik belokan terjadi *overshoot*. Namun dengan menggunakan dua pengukuran dihasilkan *steady state error* dengan nilai *settling time* kurang dari 12 s pada semua *trajectory* referensi.

Kata kunci: AGV, kontrol fuzzy, encoder, sensor magnetik, transportasi

Halaman ini sengaja dikosongkan

TRAJECTORY TRACKING OF AN AUTOMATED GUIDED VEHICLE USING FUZZY CONTROLLER ON MAGNETIC SPOT ROUTES

By : Mamat Septyan
Student Identity Number : 07111650020002
Supervisor(s) : 1. Dr. Trihastuti Agustinah, ST., MT.

ABSTRACT

Automated Guided Vehicle (AGV) is an important mode of transportation in the industry. In order to achieve flexibility to deal with repetitive tasks, a sufficient control system is needed in regulating transportation consequently this technology has been widely used as an object of research. One of the research open areas is the problem of a tracking control especially for reference trajectories with big angle changes. The limitation of movement for AGV with a nonholonomic constraints resulting the tracking on nonsmooth trajectory is difficult to apply.

This research explores the development of fuzzy Takagi-Sugeno controller and dynamic nonlinear feedback to minimize tracking error on nonsmooth trajectory. The controller is designed based on the Takagi-Sugeno fuzzy model and Parallel Distributed Compensation which its gain and stability matrix is calculated using Linear Matrix Inequality. The feedback gain is obtained from two measurement as the controller's main input and a correction factor. The calculation of the AGV dynamics model torque is obtained from auxiliary velocity input and dynamic nonlinear feedback.

Simulation results show that the controller using one measurement produces steady state tracking error with settling times value less than 15 s in all reference trajectories except for orientation errors where the settling time value is 31 s because overshoot occurs at each turn point. However, the controller using two measurement produces steady state error with settling times value less than 12 s in all reference trajectories

Key words: AGV, fuzzy control, encoder, magnetic sensor, transportation

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji bagi Allah SWT, Tuhan semesta alam.

Tesis ini disusun untuk memenuhi persyaratan guna menyelesaikan pendidikan Magister pada Bidang Studi Teknik Sistem Pengaturan, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan judul:

***“TRACKING TRAJECTORY AUTOMATED GUIDED VEHICLE
MENGUNAKAN FUZZY KONTROLER PADA RUTE
MAGNETIC SPOT”***

Terima kasih untuk semua pihak yang telah memberikan dukungan dalam pengerjaan Tesis ini. Terutama kepada Dr. Trihastuti Agustinah, S.T., M.T. selaku pembimbing atas segala bimbingan, kesabaran, dan dukungan serta motivasi hingga terselesaikannya Tesis ini. Penulis juga berterima kasih kepada teman-teman Teknik Sistem Pengaturan yang selalu memberikan kritik dan saran demi terselesaikannya Tesis ini. Penulis juga ingin menyampaikan terima kasih kepada orang tua atas segalanya dan permintaan maaf karena belum bisa membalas pengorbanan beliau berdua. Semoga buku Tesis ini dapat memberikan manfaat untuk pembaca yang tertarik pada pengembangan teknologi AGV.

Surabaya, 11 Desember 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Kontribusi	3
BAB 2 KAJIAN PUSTAKA	5
2.1 Kajian Penelitian Terkait	5
2.1.1 <i>Navigation of Automated Guided Vehicles Using Magnet Spot Guidance Method</i> [3]	5
2.1.2 <i>Robust Tracking Control for Wheeled Mobile Robot Based on Extended State Observer</i> [6]	9
2.1.3 <i>Trajectory Tracking for Nonholomic Mobile Robots Based on Extended Models</i> [11]	12
2.1.4 <i>Global Position Tracking Control of an Omni-Directional Mobile Robot using Fusion of a Magnetic Compass and Encoders</i> [12]	16
2.2 Teori Dasar	19
2.2.1 Fuzzy Takagi-Sugeno Model [13]	20
2.2.2 <i>Parallel Distributed Compensation (PDC)</i> [13]	21
2.2.3 <i>Linear Matrix Inequality (LMI)</i> [13]	22
BAB 3 METODE PENELITIAN	25
3.1 Pemodelan Kinematika dan Dinamika AGV	27

3.1.1	Struktur dan Model Kinematika Sistem	27
3.1.2	Struktur dan Model Dinamika	31
3.2	Pemodelan Magnet dan Pembacaan Sensor	32
3.3	Perancangan Kontroler	35
3.3.1	Perancangan Koreksi <i>Input Model Error</i> Kinematika	35
3.3.2	Perancangan Kontroler Fuzzy Takagi-Sugeno	38
3.3.3	Perancangan Kontroler Dinamika	42
3.3.4	Analisa <i>Steady State Error</i>	43
3.3.5	Rencana Pengujian	46
BAB 4 HASIL DAN PEMBAHASAN		49
4.1	Model Simulink dari Kontroler AGV	49
4.2	Pengujian Kontroler pada Model Kinematika	50
4.2.1	Pengujian Kontroler Fuzzy T-S pada Model Kinematika	51
4.2.2	Pengujian Perbandingan <i>Trajectory Tracking</i> antara Hasil Perancangan dengan Metode dari Kajian Pustaka	56
4.3	Pengujian Pengaruh Kecepatan Terhadap Respon Tracking	61
4.4	Pengujian Kinematika dan Dinamika Kontroler pada Model Dinamika dengan Satu Pengukuran	62
4.5	Pengujian <i>Tracking Trajectory</i> untuk Sistem dengan Dua Pengukuran	68
BAB 5 KESIMPULAN		77
DAFTAR PUSTAKA		79
LAMPIRAN		83
DAFTAR INDEX		107

DAFTAR GAMBAR

Gambar 2.1	Distribusi dari MFD untuk Pergerakan Lurus [3]	7
Gambar 2.2	Distribusi MFD untuk Pergerakan Lurus dengan Gangguan [3]	8
Gambar 2.3	Distribusi MFD untuk <i>Trajectory Tracking</i> [3]	8
Gambar 2.4	<i>Mobile Robot</i> dengan <i>Differential Drive</i> [6]	9
Gambar 2.5	<i>Error Posture</i> dari <i>Mobile Robot</i> [6]	10
Gambar 2.6	Hasil <i>Trajectory Tracking</i> dari <i>Mobile Robot</i> [6]	12
Gambar 2.7	Hasil <i>Error Trajectory Tracking</i> dari Robot [6]	12
Gambar 2.8	<i>Differential Drive Nonholonomic Robot</i> [11]	13
Gambar 2.9	<i>Trajectory</i> referensi <i>Tracking</i> berupa <i>Berzier Curve</i> [11]	14
Gambar 2.10	Hasil dari Kontrol <i>Input</i> untuk <i>Tracking</i> [11]	14
Gambar 2.11	Hasil <i>Error</i> dari <i>Trajectory Tracking</i> [11]	15
Gambar 2.12	Koordinat dan Parameter dari ODMR [12]	16
Gambar 2.13	Blok Diagram <i>Magnetic Compass</i> [12]	17
Gambar 2.14	(a) Hasil <i>Tracking</i> pada Lingkaran (b) <i>Error Tracking</i> [12]	18
Gambar 2.15	(a) Hasil <i>Tracking</i> dan <i>Error</i> tanpa Koreksi, (b) Hasil <i>Tracking</i> dan <i>Error</i> dengan Koreksi [12]	19
Gambar 3.1	(a) Diagram Blok Perancangan Sistem Kontrol, (b) <i>Dynamic Controller</i>	26
Gambar 3.2	<i>Automated Guided Vehicle</i>	27
Gambar 3.3	Model <i>Error</i> e_x , e_y , dan e_θ	28
Gambar 3.4	Perancangan Sensor <i>Magnetic</i> dan Pembacaan <i>Flux</i>	32
Gambar 3.5	Permanent Magnet	33
Gambar 3.6	Rancangan <i>Magnetic Spot</i>	33
Gambar 3.7	Distribusi MFD dengan FEMM	34
Gambar 3.8	<i>Threshold</i> untuk Pembacaan MFD	34
Gambar 3.9	Pembagian Area MFD	35
Gambar 3.10	Pembacaan Sensor Magnet (a) Saat S3 membaca (b) Saat S2 Membaca	37
Gambar 3.11	Membership function (a) Sensor 2 (b) Selisih Sensor 1 dan 3	38
Gambar 4.1	Model Simulink dari Kontroler AGV	50
Gambar 4.2	(a) <i>Trajectory Tracking</i> Garis Lurus menggunakan Model Kinematika (b) <i>Error Tracking</i>	51
Gambar 4.3	Respon Kecepatan Linier dan Angular pada Garis Lurus	52
Gambar 4.4	<i>Error Tracking</i> pada <i>Trajectory</i> Segi Delapan	53
Gambar 4.5	(a) <i>Trajectory Tracking</i> pada Persegi menggunakan Model Kinematika (b) <i>Error Tracking</i>	54
Gambar 4.6	Perbandingan <i>Tracking Error</i> pada <i>trajectory</i> Lingkaran	57
Gambar 4.7	Perbandingan Hasil <i>Tracking Error Leading</i> pada Lingkaran	57
Gambar 4.8	Perbandingan Hasil <i>Tracking Error Lateral</i> pada Lingkaran	58
Gambar 4.9	Perbandingan Hasil <i>Tracking Error Orientasi</i> pada Lingkaran	58
Gambar 4.10	Hasil <i>Tracking</i> pada Persegi [11]	59

Gambar 4.11	Hasil Perbandingan <i>Tracking Error In Leading</i> pada Persegi.....	59
Gambar 4.12	Hasil Perbandingan <i>Tracking Error Lateral</i> pada Persegi	60
Gambar 4.13	Hasil Perbandingan <i>Tracking Error Orientasi</i> pada Persegi	60
Gambar 4.14	(a) <i>Trajectory Tracking</i> Garis Lurus menggunakan Model Dinamika, (b) <i>Error Tracking</i>	62
Gambar 4.15	Kecepatan Linier dan Angular <i>Trajectory Tracking</i> Garis Lurus .	63
Gambar 4.16	Tampilan dari Matlab GUI untuk <i>Trajectory</i> Garis Lurus	63
Gambar 4.17	<i>Tracking Trajectory</i> Segidelapan menggunakan Model Dinamika, (b) <i>Error Tracking</i>	65
Gambar 4.18	Kecepatan Linier dan Angular <i>Trajectory Tracking</i> Segidelapan.	66
Gambar 4.19	(a) <i>Tracking Trajectory</i> Persegi menggunakan Model Dinamika, (b) <i>Error Tracking</i>	67
Gambar 4.20	Hasil <i>crop</i> dari Matlab GUI untuk <i>Trajectory</i> Persegi	68
Gambar 4.21	(a) <i>Tracking</i> pada <i>Trajectory</i> Segidelapan menggunakan Model Dinamika, (b) <i>Error Tracking</i>	69
Gambar 4.22	Hasil <i>crop</i> Simulasi pada Matlab GUI.....	70
Gambar 4.23	(a) <i>Trajectory Tracking</i> Persegi dengan Dua Pengukuran (b) <i>Error Tracking</i>	71
Gambar 4.24	Hasil <i>crop</i> dari Matlab GUI untuk <i>Trajectory</i> Persegi	72
Gambar 4.25	Pengujian Jarak antar <i>Magnetic Spot</i> (a) <i>Error Tracking</i> (b) Hasil <i>Tracking</i>	74
Gambar 4.26	Hasil <i>Tracking</i> dengan Gangguan Eksternal	75

DAFTAR TABEL

Tabel 2.1 Aturan Fuzzy Untuk Output AGV: <i>Steering angle dan Distance</i> [4].....	7
Tabel 4.1 Hasil Pengujian Kontroler pada Model Kinematika.....	54
Tabel 4.2 Perbandingan Settling Time Hasil dari Tesis dan Kajian Pustaka [11] pada Trajectory Lingkaran	56
Tabel 4.3 Pengujian Pengaruh Kecepatan terhadap Trajectory Tracking pada Lingkaran.....	60
Tabel 4.4 Pengujian Pengaruh Kecepatan terhadap Trajectory Tracking pada Persegi	61
Tabel 4.5 Perbandingan <i>Trajectory Tracking</i> dengan Satu Pengukuran dan Dua Pengukuran	73
Tabel 4.6 Hasil <i>Trajectory Tracking</i> menggunakan Dua Pengukuran dengan Gangguan Eksternal.....	73

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penggunaan teknologi Automated Guided Vehicle (AGV) telah meluas seiring dengan kemajuan industri salah satunya untuk transportasi bidang manufaktur [1]. Dengan banyaknya tugas berulang yang terdapat dalam proses produksi, kebutuhan transportasi menjadi cukup besar sehingga AGV banyak digunakan untuk mengatasi inefisiensi transportasi dan meminumkan kesalahan dalam *supply chain* antar mesin produksi (*material flow*) [2].

Metode kontrol untuk AGV telah mengalami perkembangan baik dalam *guidance*, *trajectory following*, maupun *trajectory tracking* [2]. Salah satunya yaitu penggunaan kontroler fuzzy Mamdani dan magnetic spot *guidance* [3]. Kontroler digunakan untuk mengendalikan pergerakan AGV pada garis lurus dimana referensi eksternal berupa magnet ditanam disepanjang *trajectory* referensi. Hasil pengukuran *magnetic flux density* (MFD) dari sensor hall-effect digunakan untuk memperbaiki *steering angle* dan *lateral distance* dengan cara *switch* dari kontroler ke metode *guidance* ketika MFD terdeteksi oleh sensor namun pada penelitian ini belum terdapat analisa kestabilan pada kontroler. Beberapa penelitian telah dilakukan untuk permasalahan kestabilan kontroler diantaranya menggunakan model *error* kinematika yang disusun berdasarkan *error tracking* dan matriks rotasi [4]. Dari model ini kemudian dirancang kontroler yang memenuhi kondisi kestabilan dari fungsi Lyapunov untuk *plant* berupa model kinematika dengan *differential drive*. Perancangan kontroler fuzzy Takagi-Sugeno berdasarkan fuzzy Takagi-Sugeno model dan disusun berdasarkan Parallel Distributed Compensation (PDC) diajukan pada [5]. Namun didalam penerapannya, model dinamika dari AGV tidak dapat diabaikan dalam pengontrolan sehingga pada [6] diajukan metode kontrol untuk model dinamika AGV menggunakan *input auxiliary velocity* dari model *error* kinematika dan menjadikannya sebagai *input* perhitungan torsi dengan menggunakan skema yang diajukan pada [7].

Penurunan model dinamika melalui pendekatan model fisik menggunakan *langrange-euler formulation* diberikan pada [8]. Untuk keperluan kontrol dan simulasi, model ditransformasi ke bentuk alternatif yang lebih sederhana dengan menghilangkan parameter yang tidak diketahui. Pada [9] dilakukan penelitian penggunaan *magnetic spot* untuk menentukan posisi dan orientasi pada pergerakan robot dimana sensor yang digunakan adalah dua sensor hall-effect 3-sumbu dengan menjaga *magnetic moment* sumbu z konstan sehingga didapatkan pengukuran MFD pada dua dimensi. Pada [10] terdapat mekanisme penentuan *threshold* untuk membedakan perubahan nilai jarak yang terukur saat mendekati atau menjauhi titik maksimum MFD. Ketika sensor hall-effect melewati titik maksimum MFD, nilai jarak akan berubah tanda sesuai dengan nilai *threshold* yang digunakan.

Pada penelitian ini digunakan kontroler fuzzy Takagi-Sugeno berdasarkan perancangan fuzzy Takagi-Sugeno model dan PDC dengan *gain* dan matriks stabilitas dihitung menggunakan LMI. Hasil dari kontroler fuzzy digunakan untuk perhitungan *auxiliary velocity* sebagai *input* kontroler dinamika. Untuk kontroler dinamika dirancang menggunakan *gain forward* dan kontrol *proportional-derivative* dengan *dynamics nonlinear feedback*.

1.2 Rumusan Masalah

Rumusan masalah dari tesis ini adalah bagaimana merancang kontroler fuzzy Takagi-Sugeno menggunakan PDC dan LMI berdasarkan model *error* kinematika 3 *state* dengan dua pengukuran dimana terdapat gangguan eksternal statis berupa besi pada area MFD.

1.3 Tujuan

Tujuan dari tesis ini yaitu dihasilkannya kontroler yang dapat meminimumkan *error* pergerakan AGV pada *trajectory* referensi dengan dua pengukuran yaitu encoder dan sensor magnet serta mampu mengatasi adanya gangguan eksternal pada MFD.

1.4 Batasan Masalah

Pada tesis ini terdapat beberapa batasan masalah yaitu MFD dalam keadaan ideal, tidak terdapat *error slipping* dan *wheel friction*.

1.5 Kontribusi

Kontribusi yang diharapkan dari hasil penelitian ini adalah sebagai berikut:

1. Didapatkan pengembangan kontroler fuzzy Takagi-Sugeno berdasarkan model *error 3 state* yang dapat meminimumkan kesalahan *tracking* AGV.
2. Didapatkan kontroler yang stabil untuk *trajectory tracking* berdasarkan perancangan Takagi-Sugeno fuzzy model dan *Parallel Distributed Compensation* (PDC) dengan perhitungan *state feedback gain* dan matriks stabilitas menggunakan *Linear Matrik Inequality* (LMI).
3. Dihasilkan pengaturan bobot dengan *degree of belief* (DOB) untuk penggunaan dua sensor pada sistem kontrol AGV.
4. Didapatkan data pengaruh kecepatan AGV terhadap *error tracking*.
5. Dihasilkan data jarak antar *magnetic spot* sehingga AGV tetap dapat melakukan *tracking*.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

Terdapat berbagai metode yang telah banyak dikembangkan untuk navigasi AGV. Permasalahan untuk metode navigasi AGV diantaranya yaitu stabilisasi model, *trajectory following*, dan *trajectory tracking* [7]. Pada tesis ini adalah dibahas tentang permasalahan *trajectory tracking*. Pada bab ini akan dijelaskan beberapa penelitian dan dasar teori yang menjadi landasan pengambilan tesis.

2.1 Kajian Penelitian Terkait

Beberapa penelitian terakhir terkait sistem AGV baik untuk kinematika, dinamika, maupun kontroler serta penelitian tentang penggunaan *magnetic spot* sebagai referensi eksternal akan dijelaskan pada sub-bab ini sebagai dasar pengambilan tesis.

2.1.1 *Navigation of Automated Guided Vehicles Using Magnet Spot Guidance Method* [3]

Pada paper ini dibahas tentang penggunaan kontroler fuzzy Mamdani dan *guidance* berupa *magnetic spot* sebagai metode navigasi AGV. *Input* dari kontroler adalah *error* pergerakan AGV dari *trajectory* referensi. Ketika sensor hall effect tidak dapat mengukur MFD, perhitungan posisi AGV dilakukan berdasarkan kecepatan angular roda kanan dan kiri yang diukur menggunakan encoder sehingga terdapat proses *switch* untuk penggunaan sensor yang ditentukan oleh dimana area AGV berada.

Permasalahan yang diangkat pada paper ini adalah bagaimana menjaga agar pergerakan dari AGV tetap pada *trajectory* referensi. Untuk menyelesaikan permasalahan tersebut digunakan dua algoritma navigasi yaitu (a) Algoritma untuk menentukan lokasi dari AGV (b) Algoritma untuk eksekusi perintah dalam meminimumkan *error* secepat mungkin. Kecepatan angular roda kanan dan kiri didapatkan dari pengukuran encoder.

Posisi dan orientasi dari AGV $(x_c(t), y_c(t), \theta_c(t))$ pada sumbu X-Y dapat dihitung sebagai berikut:

$$\begin{aligned} x_c(t) &= x_0 + \frac{r}{2} \int_0^t (\omega_L(t) + \omega_R(t)) \cos \theta_c(t) dt \\ y_c(t) &= y_0 + \frac{r}{2} \int_0^t (\omega_L(t) + \omega_R(t)) \sin \theta_c(t) dt \\ \theta_c(t) &= \theta_0 + \frac{r}{D} \int_0^t (\omega_R(t) - \omega_L(t)) dt \end{aligned} \quad (2.1)$$

dimana

$[\omega_L, \omega_R]$: kecepatan angular roda kiri dan kanan

$[x_0, y_0, \theta_0]$: Inisial posisi dan orientasi.

D : jarak antara roda kanan dan roda kiri.

r : jari-jari roda

Jenis magnet yang digunakan pada penelitian ini adalah *permanent magnet* (PM) dengan pemodelan *magnetic dipole* sebagai berikut:

$$U = \frac{M \cos \theta}{4\pi\mu_0 r^2} = \frac{Mz}{4\pi\mu_0 (x^2 + y^2 + z^2)^{\frac{3}{2}}} \quad (2.2)$$

dimana

U : *potential* dari *magnetic dipole*

x, y, z : koordinat *cathesian* dari *dipole* MFD

μ_0 : *permeability of free space*, $4\pi \times 10^{-7} \text{ H/m}$

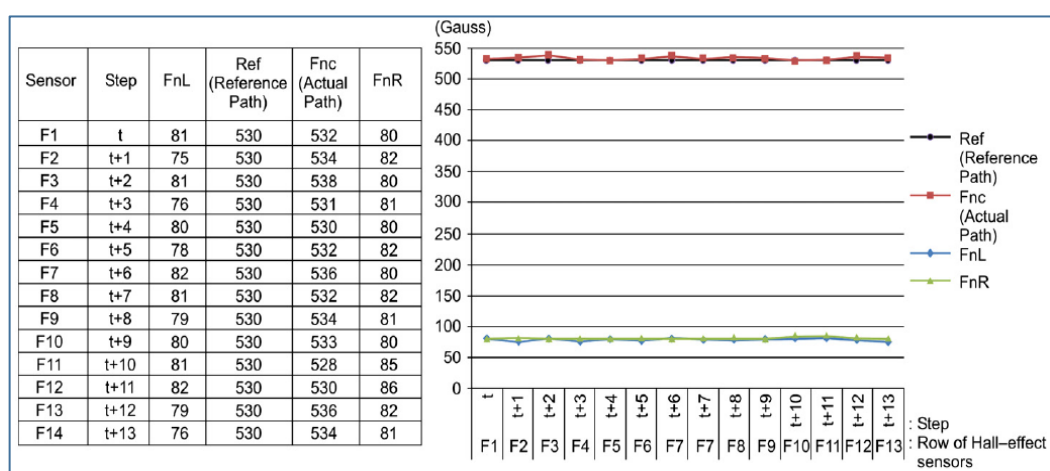
Nilai *magnitude* dari MFD (B_x, B_y , dan B_z) dapat dihitung menggunakan:

$$\begin{aligned} B_x &= \frac{3}{4} k \frac{xz}{(x^2 + y^2 + z^2)^{\frac{5}{2}}} \\ B_y &= \frac{3}{4} k \frac{yz}{(x^2 + y^2 + z^2)^{\frac{5}{2}}} \\ B_z &= \frac{1}{4} k \frac{2z^2 - x^2 - y^2}{(x^2 + y^2 + z^2)^{\frac{5}{2}}} \end{aligned} \quad (2.3)$$

dimana $k = Ma/4\pi$, a adalah koefisien magnet, B_x dan B_y adalah MFD yang diukur sensor hall-effect pada bagian depan atau belakang dan bagian kiri atau kanan dari *magnetic spot* dengan satuan *gauss*, sedangkan B_z didapatkan dari ketinggian antara *magnetic spot* dan sensor hall-effect.

Tabel 2.1 Aturan Fuzzy Untuk Output AGV: *Steering angle dan Distance* [3]

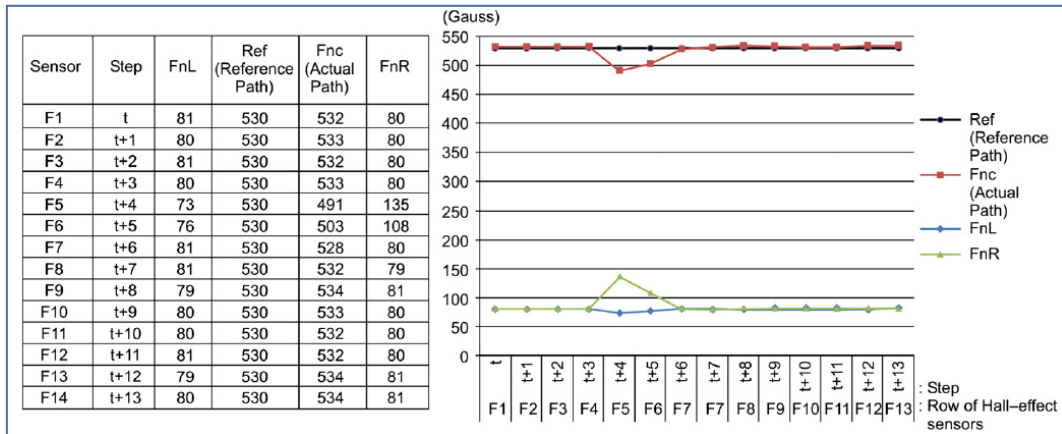
<i>Distance</i>	<i>Angle</i>				
	LL	LS	ZE	RS	RL
LF	RL	RL	RL	RS	ZE
LN	RL	RS	RS	ZE	LS
MD	RL	RS	ZE	LS	LL
RN	RS	ZE	LS	LS	LL
RF	ZE	LS	LL	LL	LL



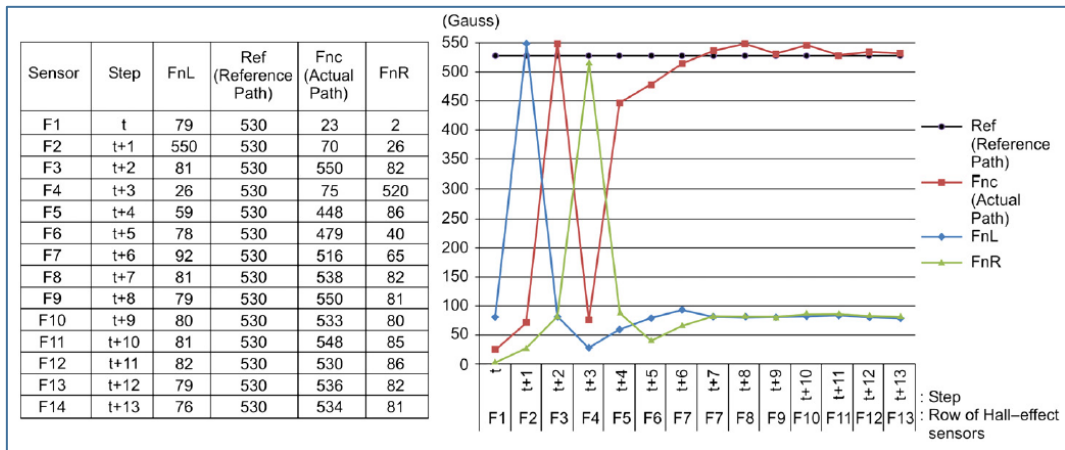
Gambar 2.1 Distribusi dari MFD untuk Pergerakan Lurus [3]

Pada penelitian ini digunakan fungsi keanggotaan *trigonometric* dan metode Mamdani MAX-MIN dengan aturan dasar sebagai berikut :

- (1) If steering angle error θ_e and lateral distance error d is Zero, then steering angle is Zero, and navigation speed v_s is “Fast Speed”.
- (2) If θ_e is “Left Small” and d is “Left Near”, θ is “Right Small” and v_s is “Right Slow Speed”.
- (3) If θ_e is “Left small” and d is “Right Far”, θ is “Left Big” and v_s is “Left Fast Speed”.

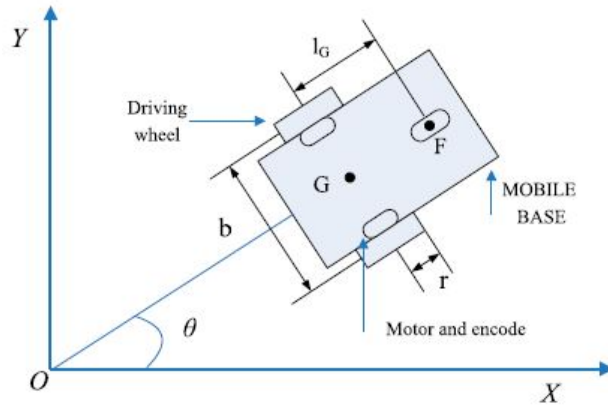


Gambar 2.2 Distribusi MFD untuk Pergerakan Lurus dengan Gangguan [3]



Gambar 2.3 Distribusi MFD untuk *Trajectory Tracking* [3]

Kelebihan dari metode pada penelitian ini yaitu terdapat koreksi *error* pergerakan AGV dari kontroler fuzzy Mamdani dengan metode *magnetic spot guidance* berdasarkan pembacaan MFD. Namun pada penelitian ini tidak ditunjukkan hasil *error posture* pergerakan AGV dari *trajectory* referensi. Parameter hasil kontrol dan *guidance* yang digunakan hanya nilai MFD yang terukur dan masih terdapat ketidaksesuaian antara data FnL dan FnR seperti yang ditunjukkan pada Gambar 2.2. Terdapat penyimpangan pada data yang ditampilkan dengan teori yang diajukan yaitu untuk *trajectory tracking* data dari distribusi *flux* menunjukkan nilai MFD yang terukur melebihi nilai MFD referensi. Belum terdapatnya analisis stabilitas untuk kontroler yang digunakan mengakibatkan tidak diketahuinya apakah kontroler sudah optimal atau belum.



Gambar 2.4 Mobile Robot dengan Differential Drive [6]

2.1.2 Robust Tracking Control for Wheeled Mobile Robot Based on Extended State Observer [6]

Pada Gambar 2.4 ditunjukkan bentuk *differential drive* dari *mobile robot* dengan persamaan kinematika dan dinamika yang dapat dinyatakan sebagai berikut:

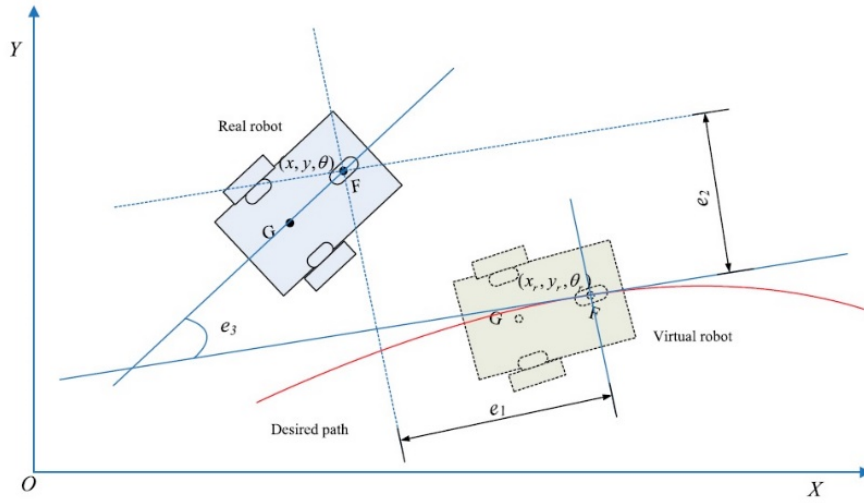
$$\begin{aligned} \dot{p} &= S(p)q \\ M(p)\dot{p} + V_m(p, \dot{p})v + F_d + \tau_d &= \tau \end{aligned} \quad (2.4)$$

dengan

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, q = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

$$S(p) = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2} & -\frac{r}{2b} \end{bmatrix}$$

dimana v dan ω merupakan kecepatan linier dan angular, r merupakan jari-jari roda, b merupakan jarak roda kanan dan roda kiri, $M(q)$ adalah matrik inersia simetri dan positif definit. Kemudian $V_m(q, \dot{q})$ adalah matrik *coriolis* dan *centripetal*, F_d adalah *surface friction*, τ adalah torsi, dan τ_d adalah gangguan eksternal.



Gambar 2.5 *Error Posture* dari Mobile Robot [6]

Konfigurasi *error tracking* dibentuk pada koordinat lokal seperti yang ditunjukkan oleh Gambar 2.5. Model *error* dapat dinyatakan sebagai berikut:

$$P_e = R(-\theta)(P_r - P)$$

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} - \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \right) \quad (2.5)$$

dimana P_e menyatakan *error* posisi dan orientasi referensi, e_x adalah *error in leading*, e_y adalah *error lateral*, e_θ adalah *error orientasi*, P_r menyatakan posisi dan orientasi referensi, dan P menyatakan posisi dan orientasi *mobile robot* serta $R(\theta)$ merupakan matriks rotasi.

Dari (2.5) dapat dibentuk persamaan differensial sebagai berikut:

$$\begin{aligned} \dot{e}_x &= \omega e_y - v + v_r \cos e_\theta \\ \dot{e}_y &= -\omega e_x + v_r \sin e_\theta \\ \dot{e}_\theta &= \omega_r - \omega \end{aligned} \quad (2.6)$$

Didefinisikan kontrol *input* dari model *error* kinematika sebagai berikut:

$$q_r = \begin{bmatrix} v_r \cos e_\theta + k_x e_x \\ \omega_r + k_y v_r e_y + k_\theta v_r \sin e_\theta \end{bmatrix} \quad (2.7)$$

dimana k_x , k_y , dan k_θ adalah *adjustable* parameter dan $q_r = (v_r, \omega_r)^T$ adalah *auxiliary velocity*, sehingga dengan menggunakan *output* dinamika robot (v) akan didapatkan *error* dari model dinamika sebagai berikut:

$$e = q_r - q \quad (2.8)$$

Berdasarkan (2.4) dapat dibentuk *input* kontroler sebagai berikut:

$$\dot{q}_0 = -M^{-1}(p)[V_m(p, \dot{p})q_0 + F_d + \tau_d] + M^{-1}(p)\tau \quad (2.9)$$

Didefinisikan $u = M^{-1}(p)\tau$ dan $q_t = -M^{-1}(p)[V_m(p, \dot{p})q_0 + F_d + \tau_d]$, sehingga (2.9) dapat ditulis menjadi:

$$\dot{q}_0 = q_t + u \quad (2.10)$$

Turunan pertama dari variabel q_t terhadap waktu dinyatakan sebagai h , sehingga dapat dibentuk model *extended* sebagai berikut:

$$\begin{aligned} \dot{q}_0 &= q_t + u \\ \dot{q}_t &= h \end{aligned} \quad (2.11)$$

Model *extended state observer* sistem (2.11) dirancang sebagai berikut:

$$\begin{aligned} \dot{\hat{q}}_0 &= q_t - z_0 r_0 + u \\ \dot{\hat{q}}_t &= -z_t fal(r_0, \alpha, \delta) \end{aligned} \quad (2.12)$$

dengan

$$fal(r_0, \alpha, \delta) = |r_0|^\alpha sign(r_0)$$

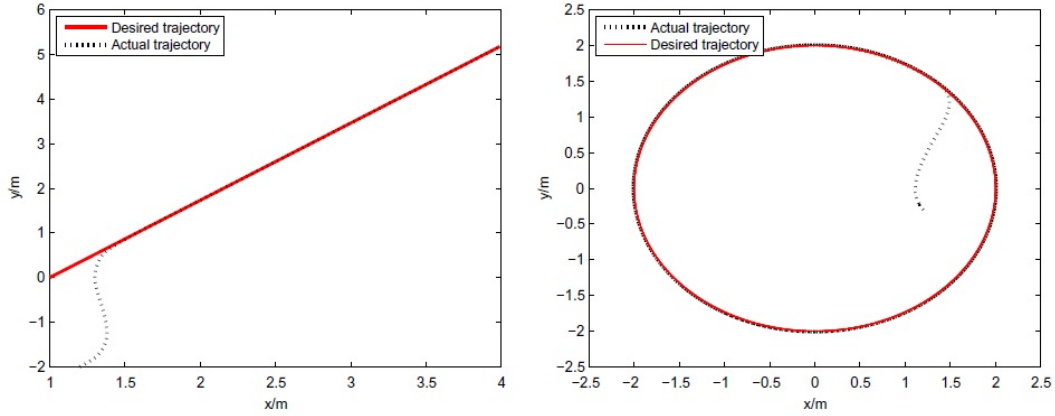
dimana z_0 dan z_t merupakan *adjustable* parameter, $fal(r_0, \alpha, \delta)$ merupakan fungsi nonlinear, α dan δ adalah konstanta, $|r_0| > \delta$, dan *error* sistem dinyatakan dalam bentuk:

$$\begin{aligned} \dot{r}_0 &= r_t - z_0 r_0 \\ \dot{r}_t &= -z_t fal(r_0, \alpha, \delta) - h \end{aligned} \quad (2.13)$$

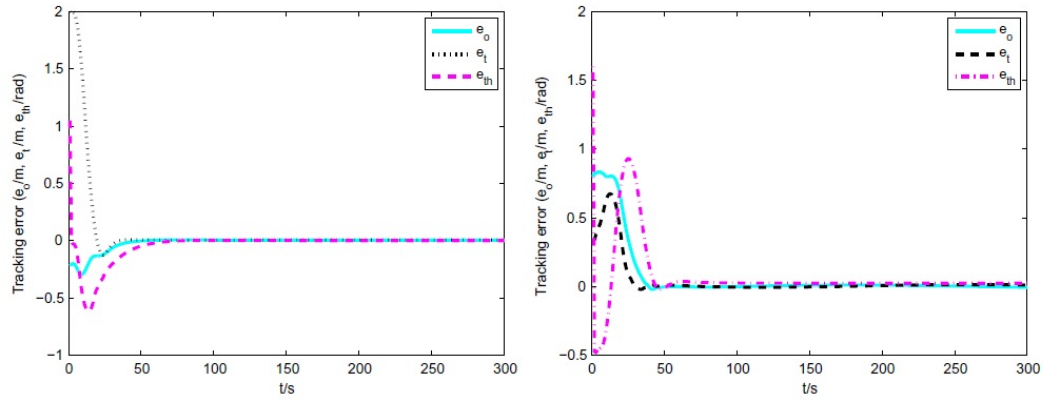
dimana $r_0 = x_0 - \hat{x}_0$ dan $r_t = x_t - \hat{x}_t$ merupakan *extended state error*.

Dengan mengatur nilai z_0 dan z_t serta menentukan fungsi nonlinear $fal(r_0, \alpha, \delta)$ menurut penelitian ini akan didapatkan nilai state \hat{x}_0 dan \hat{x}_t yang mendekati nilai x_0 dan x_t . Hasil dari penelitian ini ditunjukkan pada Gambar 2.6.

Dari hasil pengujian ditunjukkan bahwa permasalahan *trajectory tracking* dapat diselesaikan dengan model *error* kinematika walaupun *steady state error* termasuk lambat untuk kecepatan yang rendah yaitu 60 s. Perancangan *adjustable* parameter (k_x, k_y, k_θ) dilakukan berdasarkan fungsi Lyapunov namun tidak ditunjukkan nilai *steady state* yang dihasilkan. Kemudian batasan minimum atau maksimum dari *adjustable parameter* tidak diberikan sehingga belum dapat ditentukan apakah performa yang ditampilkan sudah optimal karena dengan *error* inisial yang kecil didapatkan nilai *overshoot error* dan *settling time* yang besar.



Gambar 2.6 Hasil *Trajectory Tracking* dari *Mobile Robot* [6]



Gambar 2.7 Hasil *Error Trajectory Tracking* dari Robot [6]

2.1.3 *Trajectory Tracking for Nonholomic Mobile Robots Based on Extended Models* [11]

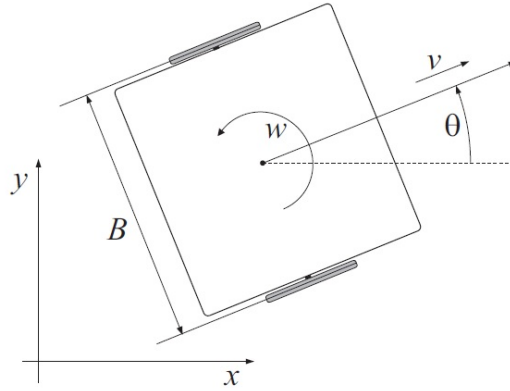
Persamaan model *error* kinematika dari *mobile robot* dapat dinyatakan dalam bentuk sebagai berikut:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \cos e_\theta & 0 \\ \sin e_\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u \quad (2.14)$$

dimana $u = [v, \omega]^T$ adalah kecepatan linier dan angular, e_x adalah *error in leading*, e_y adalah *error lateral*, e_θ adalah *error orientasi*.

Model *extended* dari (2.14) dibentuk melalui penambahan *state* pada model *error* kinematika dengan mendefinisikan *state* baru yaitu:

$$\begin{aligned} s(t) &= \sin(\theta(t)) \\ c(t) &= \cos(\theta(t)) \end{aligned} \quad (2.15)$$



Gambar 2.8 *Differential Drive Nonholonomic Robot* [11]

Turunan pertama dari (2.15) terhadap waktu adalah sebagai berikut:

$$\begin{aligned}\dot{s}(t) &= \cos(\theta(t))\dot{\theta}(t) = c(t)\omega(t) \\ \dot{c}(t) &= -\sin(\theta(t))\dot{\theta}(t) = -s(t)\omega(t)\end{aligned}\quad (2.16)$$

Didefinisikan *state error* baru berdasarkan (2.16), sehingga model *extended* dari model *error* kinematika didapatkan:

$$\begin{aligned}e_x &= c(x_r - x) + s(y_r - y) \\ e_y &= -s(x_r - x) + c(y_r - y) \\ e_s &= \sin(\theta_r - \theta) = s_r c - c_r s \\ e_c &= \cos(\theta_r - \theta) = c_r c + s_r s\end{aligned}\quad (2.17)$$

Model *error* kinematika pada (2.14) diganti dengan model baru sebagai berikut:

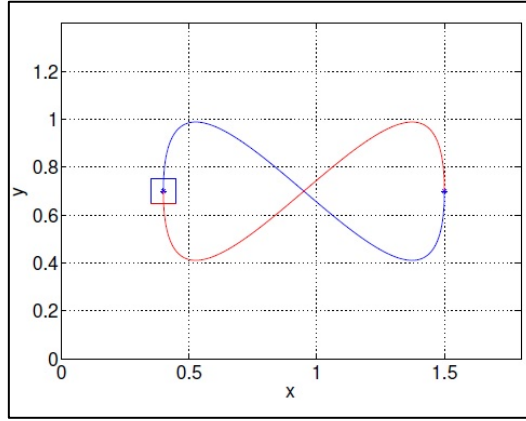
$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_s \\ \dot{e}_c \end{bmatrix} = \begin{bmatrix} e_c & 0 \\ e_s & 0 \\ 0 & e_c \\ 0 & -e_s \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -e_c \\ 0 & e_s \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}\quad (2.18)$$

Pada (2.18) *state* e_c merupakan bagian dari sistem yang *uncontrollable* sehingga sistem dibagi menjadi dua sebagai berikut:

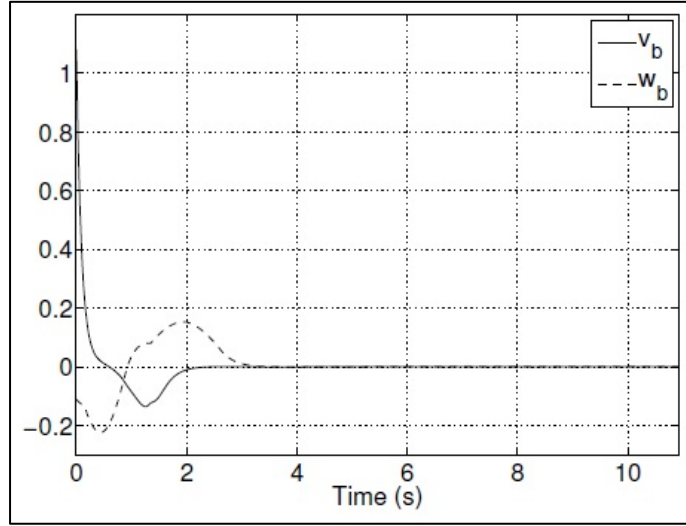
$$\dot{e} = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_s \end{bmatrix} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -e_c \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}\quad (2.19)$$

$$\dot{e}_c = e_s \omega_b \quad (2.20)$$

Kontroler akan dirancang berdasarkan sistem (2.19) sedangkan sistem (2.20) akan digunakan untuk pengecekan apakah nilai e_c konvergen menuju nilai 1 atau tidak.



Gambar 2.9 *Trajectory* referensi *Tracking* berupa *Berzier Curve* [11]



Gambar 2.10 Hasil dari Kontrol *Input* untuk *Tracking* [11]

Fuzzy T-S Model dapat ditulis sebagai berikut:

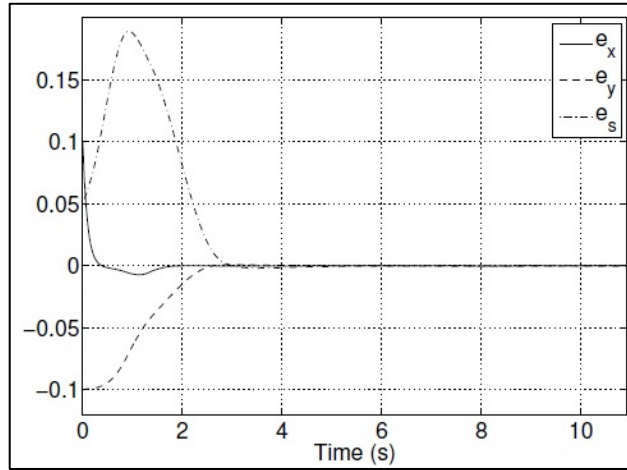
$$\dot{e}(t) = \sum_{i=1}^r h_i(z(t))(A_i e(t) + B_i u(t)) \quad (2.21)$$

Variabel *premise* (z) yang digunakan adalah:

$$z(t) = [w_r(t) \quad v_r(t) \quad e_y(t) \quad e_x(t) \quad e_c(t)] \quad (2.22)$$

Matriks linearisasi dirancang sebagai berikut:

$$A_i = \begin{bmatrix} 0 & \varepsilon_i^1 & 0 \\ -\varepsilon_i^1 & 0 & \varepsilon_i^2 \\ 0 & 0 & 0 \end{bmatrix}, B_i = \begin{bmatrix} -1 & \varepsilon_i^3 \\ 0 & -\varepsilon_i^4 \\ 0 & -\varepsilon_i^5 \end{bmatrix}, i = 1 \dots r \quad (2.23)$$



Gambar 2.11 Hasil *Error* dari *Trajectory Tracking* [11]

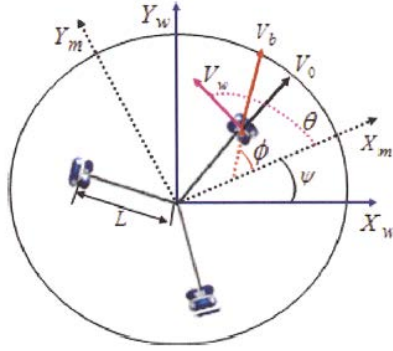
Pengaturan bobot kriteria (w) adalah sebagai berikut:

$$w_i(z) = \prod_{j=1}^5 M_{ij}(z_j), \quad i = 1, 2, \dots, 32 \quad (2.24)$$

$$w_1^j(z_j) = \frac{z_j - \underline{z}_j}{\bar{z}_j - \underline{z}_j}, w_0^j(z_j) = 1 - w_1^j(z_j), j = 1, 2, \dots, 5 \quad (2.25)$$

dimana i adalah jumlah aturan, \bar{z}_j dan \underline{z}_j adalah nilai maksimum dan minimum *premise*.

Hasil kecepatan linear dan angular *mobile robot* yang didapatkan dari kontroler fuzzy T-S untuk sistem (2.19) ditunjukkan pada Gambar 2.10 dan untuk hasil *error tracking* ditunjukkan pada Gambar 2.11. Dari hasil pengujian terlihat bahwa kontrol *input* dan *error tracking* konvergen menuju titik 0 sehingga *mobile robot* bergerak pada *trajectory* sesuai dengan *input* referensi v_r dan ω_r . Akan tetapi pengujian hanya dilakukan pada *smooth trajectory* dengan perubahan v_r dan w_r yang kecil dan untuk nilai inisial *error* yang kecil didapatkan hasil *error* orientasi yang cukup besar yaitu 0.18. Dari Gambar 2.11 terlihat *state error* baru e_s mencapai nilai 0 pada 5 s namun *state error* e_c sudah bernilai 1 pada 2 s sehingga terdapat penyimpangan data dari hasil kedua *state error* baru karena jumlah keduanya harus bernilai 1.



Gambar 2.12 Koordinat dan Parameter dari ODMR [12]

2.1.4 Global Position Tracking Control of an Omni-Directional Mobile Robot using Fusion of a Magnetic Compass and Encoders [12]

Pada penelitian ini digunakan *magnetic compass* untuk memperbaiki *error* orientasi dari pengukuran encoder untuk tracking *Omni-directional mobile robot* (ODMR). Terdapat beberapa kelemahan untuk penggunaan encoder dalam *dead-reckoning* seperti ketika roda yang tergelincir atau terjadi kesalahan perhitungan encoder yang dapat menyebabkan meningkatkan *error tracking* ODMR. Kontroler yang digunakan pada penelitian adalah kontroler PID.

Koordinat $Z_m(X_m, Y_m)$ menyatakan koordinat lokal dari ODMR dan $Z_w(X_w, Y_w)$ menyatakan koordinat global seperti yang ditunjukkan pada Gambar 2.12. Kecepatan translasi dan rotasi dari ODMR dapat ditulis sebagai berikut:

$$V_{wi} = V_{bi} + V_0 \quad (2.26)$$

dimana i menyatakan nomor roda (1,2,3), V_w merupakan kecepatan roda, V_b merupakan kecepatan translasi, dan V_0 merupakan kecepatan rotasi.

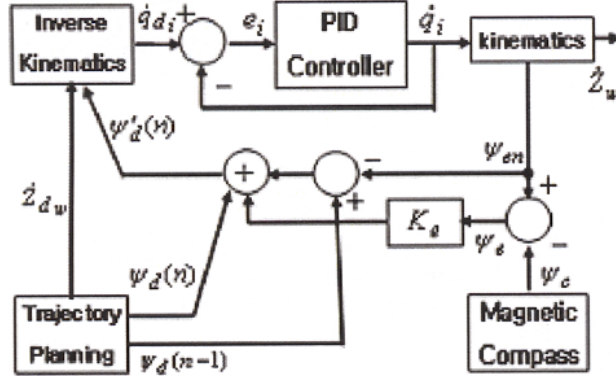
Kecepatan translasi dari roda 1 memenuhi persamaan berikut:

$$V_{b,1} = \sin(\psi)\dot{x}_w + \cos(\psi)\dot{y}_w \quad (2.27)$$

dimana ψ merupakan sudut rotasi.

Karena perbedaan sudut antar roda adalah sebesar 120° , misal diambil sumbu X_w sebagai titik awal dan sudut dihitung searah jarum jam didapatkan nilai α (sudut masing-masing sudut roda) yaitu $\alpha_1 = 0^\circ$, $\alpha_2 = 120^\circ$, dan $\alpha_3 = 240^\circ$. Kemudian dengan menentukan titik offset masing-masing roda yaitu $\psi + \alpha_i$, persamaan untuk v_i dapat ditulis sebagai berikut:

$$V_{b,i} = -\sin(\theta + \alpha_i)\dot{x}_w + \cos(\theta + \alpha_i)\dot{y}_w \quad (2.28)$$



Gambar 2.13 Blok Diagram *Magnetic Compass* [12]

Ketika ODMR melakukan gerak rotasi, V_{wi} harus memenuhi persamaan berikut:

$$V_0 = L\dot{\psi} \quad (2.29)$$

dimana R merupakan jarak roda dari titik pusat ODMR.

Kemudian dari (2.26) dan (2.27) didapatkan:

$$V_{b,1} = -\sin(\psi + \alpha_i)\dot{x}_w + \cos(\psi + \alpha_i)\dot{y}_w + L\dot{\psi} \quad (2.30)$$

Hubungan antara kecepatan translasi dan kecepatan angular (ϕ) adalah:

$$V_{wi} = r\dot{\phi}_i \quad (2.31)$$

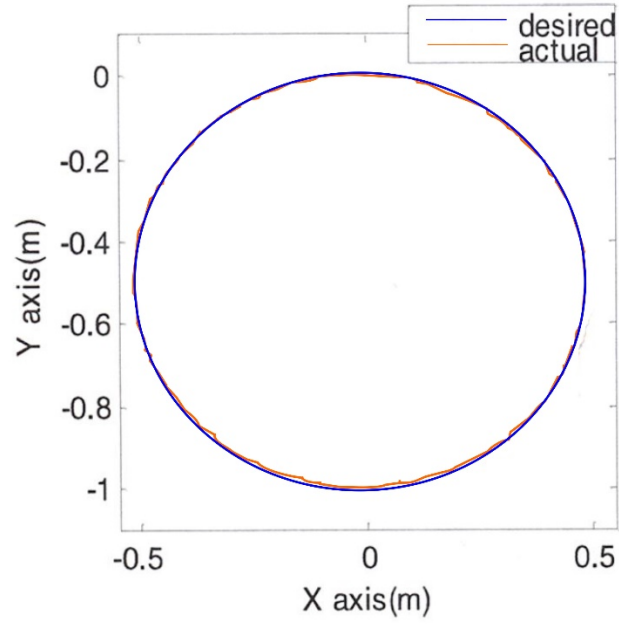
dimana r merupakan jari-jari roda.

Dari (2.28) dan (2.29) dapat ditentukan persamaan untuk kecepatan angular sebagai berikut:

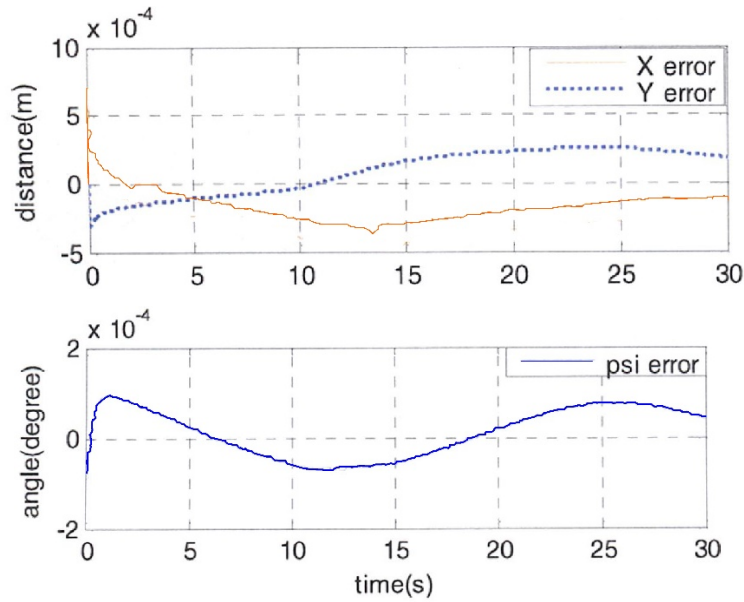
$$\dot{\phi}_i = \frac{1}{r}(-\sin(\psi + \alpha_i)\dot{x}_w + \cos(\psi + \alpha_i)\dot{y}_w + L\dot{\psi}) \quad (2.32)$$

Dengan demikian untuk persamaan kinematika dari OMDR didapatkan sebagai berikut:

$$\dot{q} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\psi) & \cos(\psi) & L \\ -\sin(\psi + \alpha_2) & \cos(\psi + \alpha_2) & L \\ -\sin(\psi + \alpha_3) & \cos(\psi + \alpha_3) & L \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\psi} \end{bmatrix} \quad (2.33)$$



(a)



(b)

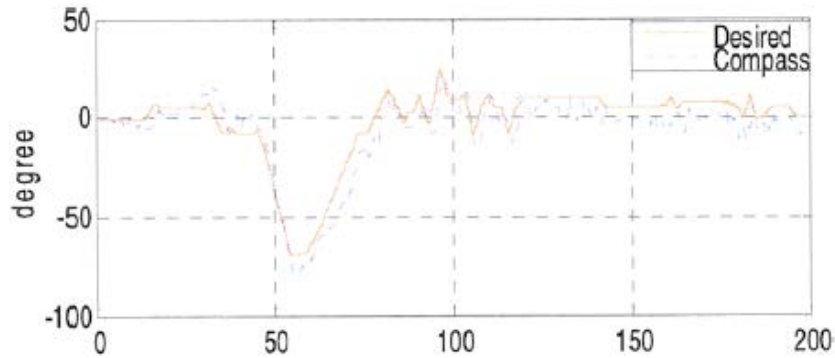
Gambar 2.14 (a) Hasil Tracking pada Lingkaran (b) *Error Tracking* [12]

Blok diagram dari koreksi dengan menggunakan *magnetic compass* ditunjukkan pada gambar 2.12. Koreksi dari sudut *heading* ODMR adalah sebagai berikut:

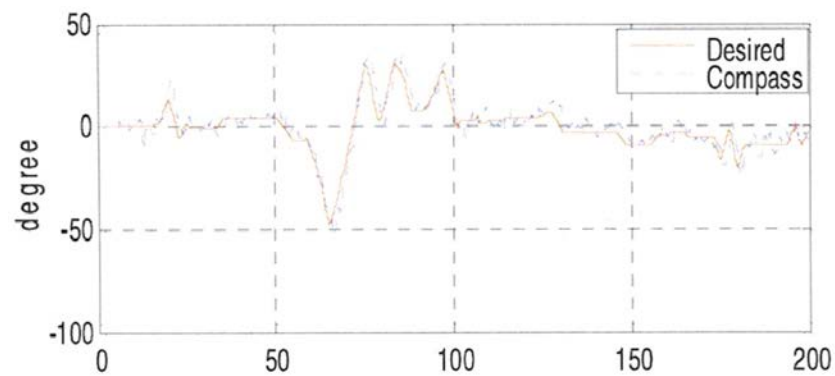
$$\psi'_a(n) = \psi_a(n) + (\psi_a(n-1) - \psi_{en}(n-1)) + K_e \psi_e(n-1) \quad (2.34)$$

dimana ψ'_a merupakan kecepatan roda setelah dikoreksi, n adalah waktu iterasi, $n-1$ adalah waktu sebelumnya, dan K_e adalah *gain* faktor yang diatur sebesar 0.8.

Hasil tracking pada lingkaran dengan menggunakan koreksi dari magnetic compass ditunjukkan Gambar 2.14 sedangkan pada Gambar 2.15 ditunjukkan hasil tracking dengan lintasan berupa titik acak yang dikirimkan lewat komunikasi *wireless* ke ODMR baik dengan koreksi maupun tidak.



(a)



(b)

Gambar 2.15 Hasil *Tracking* (a) tanpa Koreksi, (b) dengan Koreksi [12]

Dari hasil *tracking* dengan koreksi *magnetic compass* pada sudut rotasi, ψ , didapatkan *error* yang lebih kecil dibandingkan tanpa koreksi walaupun nilai *error* masih besar akibat adanya gangguan eksternal yang tidak diketahui pada pengukuran *magnetic compass* berupa vibrasi atau pengaruh dari komponen elektrik seperti driver motor.

2.2 Teori Dasar

Pada bab ini akan dijelaskan beberapa landasan teori yang digunakan untuk mendukung perumusan dan penyelesaian masalah baik untuk model sistem, kontroler, maupun analisis kondisi kestabilan kontroler.

2.2.1 Fuzzy Takagi-Sugeno Model [13]

Sistem nonlinear dapat didekati dengan menggunakan beberapa model linier melalui linearisasi pada titik kerja tertentu. Untuk fuzzy T-S model dapat dirancang menggunakan dua pendekatan yaitu identifikasi menggunakan data *input-output* atau penurunan dari persamaan sistem nonlinear.

Pada tesis ini akan digunakan pendekatan yang kedua dimana sistem nonlinear dari model *error tracking* kinematika AGV akan dilinearisasi pada titik kerja tertentu untuk mendapatkan model yang sesuai dalam penyelesaian permasalahan *trajectory tracking*.

Aturan *plant* dari fuzzy T-S model dapat ditulis sebagai berikut:

$$\begin{aligned} & \text{if } z_1(t) \text{ is } M_{i1} \text{ and } z_2(t) \text{ is } M_{i2} \dots z_p \text{ is } M_{ij} \\ & \text{then } \dot{x}(t) = A_i x(t) + B_i u(t) \quad ; i = 1, 2, \dots, r \\ & y(t) = C_i x(t) \end{aligned} \quad (2.35)$$

dimana M_{ij} menyatakan himpunan fuzzy, z_i, \dots, z_p menyatakan variabel *premise*, r adalah jumlah aturan, $x(t)$ adalah vektor *state*, $y(t)$ adalah vektor *output*, dan $u(t)$ adalah variabel *input*. Matriks A , B dan C adalah matriks transformasi *state*, *input*, dan pengukuran.

Output akhir dari fuzzy model adalah sebagai berikut:

$$\dot{x}(t) = \frac{\sum_{i=1}^r w_i(z(t)) [A_i x(t) + B_i u(t)]}{\sum_{i=1}^r w_i(z(t))} \quad (2.36)$$

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\} \\ y(t) &= \frac{\sum_{i=1}^r w_i(z(t)) C_i x(t)}{\sum_{i=1}^r w_i(z(t))} \end{aligned} \quad (2.37)$$

$$y(t) = \sum_{i=1}^r h_i(z(t)) C_i x(t)$$

dengan

$$z(t) = [z_1(t), z_2(t), \dots, z_p(t)]$$

$$w_i(z(t)) = \prod_{j=1}^n M_{ij}(z_j(t))$$

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))}$$

Pada tahap inferensi digunakan operator *product* AND dan pada tahap defuzzifikasi digunakan teknik *weighted average* sebagai berikut:

$$m_i(z) = \frac{\sum_{i=1}^r b_i \mu_i(z)}{\sum_{i=1}^r \mu_i(z)} \quad (2.38)$$

dimana $\mu_i(z)$ adalah nilai bobot tiap aturan fuzzy dan b_i adalah titik tengah fungsi keanggotaan.

2.2.2 Parallel Distributed Compensation (PDC) [13]

Untuk mempermudah analisis kestabilan dari kontroler fuzzy T-S digunakan PDC sebagai prosedur penyusunan. Aturan dari PDC ditentukan berdasarkan aturan *plant* dari fuzzy T-S model dengan fungsi keanggotaan yang sama. Aturan *plant* dari fuzzy T-S model dirancang menggunakan PDC sebagai berikut:

$$\begin{aligned} & \text{if } z_1(t) \text{ is } M_{i1} \text{ and } z_2(t) \text{ is } M_{i2} \dots z_p \text{ is } M_{ij} \\ & \text{Then } u(t) = -F_i x(t) \quad ; i = 1, 2, \dots, r \end{aligned} \quad (2.39)$$

dimana F_i adalah *gain feedback*.

Penentuan *gain feedback* dapat dilakukan dengan beragam cara salah satunya yaitu dengan metode kontrol linier seperti *pole placement*.

Secara keseluruhan kontroler fuzzy dapat ditulis sebagai berikut:

$$u(t) = -\frac{\sum_{i=1}^r w_i(z(t)) F_i x(t)}{\sum_{i=1}^r w_i(z(t))} = -\sum_{i=1}^r h_i(z(t)) F_i x(t) \quad (2.40)$$

Dengan substitusi (2.40) ke (2.36) diperoleh model lup tertutup dari sistem sebagai berikut:

$$\dot{x}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t))h_j(z(t))\{A_i - B_i F_j(t)\}x(t) \quad (2.41)$$

Berdasarkan (2.36) dan (2.40) dapat dibentuk LMI yang digunakan untuk penentuan *gain* kontroler fuzzy T-S.

2.2.3 Linear Matrix Inequality (LMI) [13]

LMI merupakan matriks dengan bentuk:

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.42)$$

dimana x_i adalah matriks variabel, F_i merupakan matriks simetri $F_i^T \in R^{n \times n}$ yang akan ditentukan. Simbol pertidaksamaan menyatakan bahwa F harus bernilai positif definit.

Fungsi Lyapunov untuk mencari kondisi kestabilan dari sebuah sistem lup terbuka adalah sebagai berikut:

$$A^T P + P A < 0 \quad (2.43)$$

dimana A adalah matriks transformasi *state* dan P adalah matriks stabilitas.

Sistem lup terbuka dari (2.36) adalah:

$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t))A_i x(t) \quad (2.44)$$

Berdasarkan (2.43) titik *equilibrium* dari sistem (2.44) adalah stabil asimtotik apabila terdapat sebuah matriks positif definit P yang memenuhi:

$$A_i^T P + P A_i < 0, \quad i = 1, 2, \dots, r \quad (2.45)$$

Kondisi kestabilan dari (2.42) dapat diuji menggunakan fungsi Lyapunov dalam bentuk kuadratik pada persamaan berikut:

$$V(x) = x(t)^T P x(t) \quad (2.46)$$

Jika terdapat sebuah matriks positif definit P yang memenuhi (2.44) sistem pada (2.44) adalah *quadratically stable*.

Titik *equilibrium* dari sistem lup tertutup (2.41) adalah stabil asimtotik global jika terdapat matrik positif definit P yang memenuhi:

$$(A_i - B_i F_j)^T P + P(A_i - B_i F_j) < 0 \quad (2.47)$$

dengan $h_i(z(t)) \cdot h_i(z(t)) \neq 0, \forall t, i, j = 1, 2, \dots, r$

Didefinisikan variabel G_{ij} sehingga (2.47) dapat ditulis menjadi:

$$\begin{aligned} \dot{x}(t) = & \left[\sum_{i=1}^r h_i(z(t)) h_i(z(t)) \{A_i - B_i F_i(t)\} x(t) \right. \\ & \left. + 2 \sum_{i=1}^r \sum_{i < j} h_i(z(t)) h_j(z(t)) G_{ij} x(t) \right] \end{aligned} \quad (2.48)$$

dengan

$$G_{ij} = \frac{\{A_i - B_i F_j\} + \{A_j - B_j F_i\}}{2}, i < j, h_i \cap h_j \neq \emptyset \quad (2.49)$$

Dengan demikian permasalahan perancangan kontrol adalah bagaimana menentukan F_i yang memenuhi kondisi (2.47) dan (2.48)

Halaman ini sengaja dikosongkan

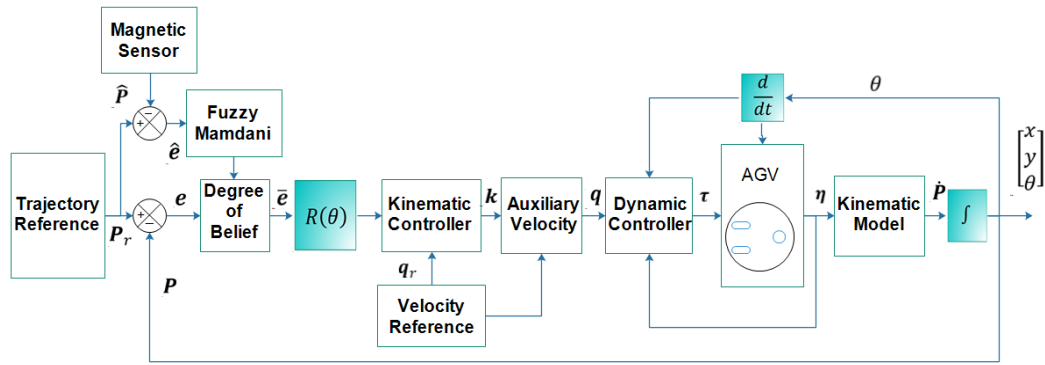
BAB 3

METODE PENELITIAN

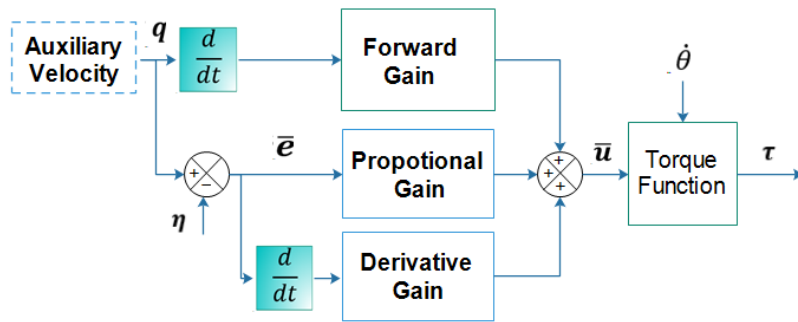
Hasil dari perancangan metode kontrol pada tesis ini ditunjukkan pada Gambar 3.1. Pada bab ini bahasan akan dibagi menjadi tiga bagian yaitu (1) pemodelan AGV baik secara kinematika maupun dinamika, (2) tahap kedua pemodelan sensor magnet dengan menggunakan Finite Element Analysis (FEA), dan (3) tahap ketiga perancangan kontroler.

Pada tesis ini digunakan encoder sebagai pengukur pergerakan AGV secara kontinu dan sensor hall-effect sebagai pengukur MFD untuk menentukan koreksi *error* pergerakan AGV pada area *magnetic spot*. Pengukuran encoder dilakukan berdasarkan kecepatan angular roda kanan dan kiri AGV sedangkan pengukuran sensor magnet dilakukan berdasarkan pembacaan MFD dari *magnetic spot* yang ditanam pada *trajectory* referensi. Pergerakan AGV dibandingkan dengan *trajectory* referensi dan *error* yang dihasilkan dikalikan dengan matriks rotasi sehingga didapatkan *error posture*. Kemudian *error* pergerakan AGV akan dikoreksi menggunakan referensi eksternal berupa *magnetic spot*. *Trajectory* referensi yang dirancang pada tesis ini terdiri dari beberapa macam dengan sudut antara 0.7854 dan 1.508 radian kecuali untuk garis lurus. Untuk pemilihan bobot dari kedua *error* digunakan DOB dengan mempertimbangkan jarak pergerakan.

Hasil *error posture* dari AGV digunakan untuk menyusun model *error tracking* kinematika 3 *state*. Dari model ini kemudian dibentuk fuzzy Takagi-Sugeno model dan dirancang kontroler fuzzy Takagi-Sugeno. Kontroler disusun berdasarkan *parallel distributed compensation* (PDC) dengan menggunakan prinsip kestabilan Lyapunov dalam bentuk LMI. Hasil keluaran kontroler fuzzy Takagi-Sugeno digunakan untuk menghitung *input auxiliary velocity* yang digunakan sebagai *input* kontroler dinamika. Namun karena keluaran dari *input auxiliary velocity* dalam bentuk kecepatan linier dan angular AGV, sedangkan model dinamika menggunakan kecepatan angular roda kanan dan roda kiri, sehingga dibutuhkan konversi. Model dinamika dari AGV disusun berdasarkan pendekatan model fisik menggunakan *Langrange-Euler formulation*.



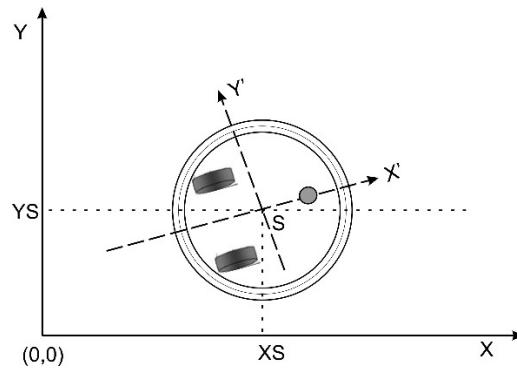
(a)



(b)

Gambar 3.1 (a) Diagram Blok Perancangan Sistem Kontrol, (b) *Dynamic Controller*

Sistem kontrol dirancang menggunakan kontroler kinematika dan dinamika seperti yang ditunjukkan oleh Gambar 3.1. Kontroler kinematika dibangun menggunakan kontroler fuzzy T-S dengan *feedback* dari dua pengukuran. Kontroler dinamika dibangun menggunakan *forward gain* dan kontroler *proportional-derivative*. Dari *input auxiliary velocity* dan *dynamic nonlinear feedback* akan didapatkan nilai torsi sebagai sinyal kontrol pada AGV. Untuk perancangan koreksi *error tracking* digunakan pengukuran dari MFD kemudian bobot dari pengukuran ini diatur dengan menggunakan DOB. Kemudian fuzzy Mamdani digunakan untuk mendapatkan nilai *feedback* yang tepat dari pengukuran MFD karena terdapat gangguan eksternal berupa besi. Untuk referensi yang digunakan yaitu posisi dan orientasi (x_r, y_r, θ_r) dan kecepatan linear dan angular (v_r, ω_r) .



Gambar 3.2 *Automated Guided Vehicle*

3.1 Pemodelan Kinematika dan Dinamika AGV

Pada sub-bab ini akan dijelaskan mengenai pemodelan dari AGV baik kinematika maupun dinamika. Model AGV berbentuk *differential drive* yang terdiri dari dua buah roda independen. Bidang *frame* yang digunakan yaitu dua dimensi sumbu X dan Y dengan koordinat yang telah ditentukan.

3.1.1 Struktur dan Model Kinematika Sistem

Pengaturan arah pergerakan AGV dapat dilakukan dengan memanfaatkan perbedaan kecepatan antara roda kiri dan roda kanan. Persamaan untuk kecepatan linier dan angular yang diperoleh dari hasil pengukuran kecepatan roda dan kiri [8] adalah:

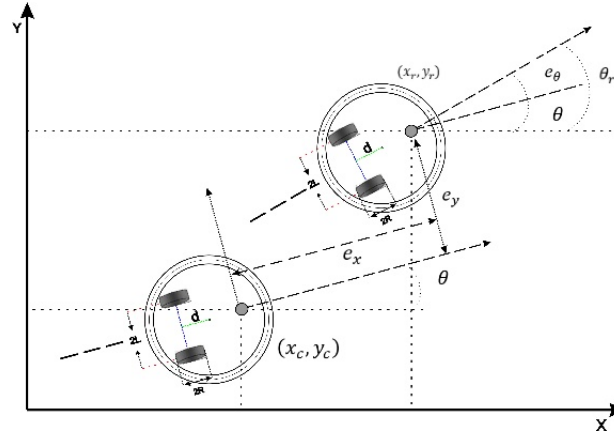
$$\begin{aligned} v &= \frac{v_R + v_L}{2} = R \frac{(\omega_R + \omega_L)}{2} \\ \omega &= \frac{v_R - v_L}{2L} = R \frac{(\omega_R - \omega_L)}{2} \end{aligned} \quad (3.1)$$

Nonholonomic *constraints* (*no lateral slip motion*) dan model kinematika untuk sumbu X dan Y ditunjukkan oleh persamaan berikut [8]:

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.2)$$

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \quad (3.3)$$

dimana p merupakan posisi dan orientasi AGV dan (v, ω) merupakan kecepatan linier dan angular. Dengan demikian dapat dihitung posisi dan orientasi pergerakan AGV menggunakan (2.1).



Gambar 3.3 Model *Error* e_x , e_y , dan e_θ

Pada koordinat global 0-XY tidak dapat ditentukan *error posture* AGV sehingga digunakan koordinat lokal S-X'Y' dengan mentransformasi posisi dan orientasi dari koordinat global ke koordinat lokal seperti yang ditunjukkan oleh Gambar 3.3. Pada koordinat lokal dapat ditentukan vektor *error tracking* seperti pada (2.5) dan (2.6) dengan menggunakan (3.3) sebagai berikut:

$$\begin{aligned}
 \dot{e}_x &= \frac{d}{dt} ((x_r - x) \cos \theta + (y_r - y) \sin \theta) \\
 &= (\dot{x}_r - \dot{x}) \cos \theta + (\dot{y}_r - \dot{y}) \sin \theta - (x_r - x) \dot{\theta} \sin \theta + (y_r - y) \dot{\theta} \cos \theta \\
 &= \dot{x}_r \cos \theta + \dot{y}_r \sin \theta - (\dot{x} \cos \theta + \dot{y} \sin \theta) + \dot{\theta} ((x - x_r) \sin \theta + (y_r - y) \cos \theta) \\
 &= \dot{x}_r \cos \theta + \dot{y}_r \sin \theta - v + \omega e_y \\
 &= \dot{x}_r \cos(\theta_r - e_\theta) + \dot{y}_r \sin(\theta_r - e_\theta) - v + \omega e_y \\
 &= \dot{x}_r (\cos \theta_r \cos e_\theta + \sin \theta_r \sin e_\theta) + \dot{y}_r (\sin \theta_r \cos e_\theta - \cos \theta_r \sin e_\theta) - v + \omega e_y \\
 &= \cos e_\theta (\dot{x}_r \cos \theta_r + \dot{y}_r \sin \theta_r) + \sin e_\theta (\dot{x}_r \sin \theta_r - \dot{y}_r \cos \theta_r) - v + \omega e_y \\
 &= v_r \cos e_\theta - v + \omega e_y \\
 \dot{e}_y &= \frac{d}{dt} (-(x_r - x) \sin \theta + (y_r - y) \cos \theta) \\
 &= -\dot{x}_r \sin \theta + \dot{y}_r \cos \theta + \dot{x} \sin \theta - \dot{y} \cos \theta - \dot{\theta} ((x_r - x) \sin \theta + (y_r - y) \cos \theta) \\
 &= -\dot{x}_r \sin \theta + \dot{y}_r \cos \theta + \dot{x} \sin \theta - \dot{y} \cos \theta - \omega e_x \\
 &= -\dot{x}_r \sin(\theta_r - e_\theta) + \dot{y}_r \cos(\theta_r - e_\theta) - \omega e_x \\
 &= -\dot{x}_r (\sin \theta_r \cos e_\theta - \cos \theta_r \sin e_\theta) + \dot{y}_r (\cos \theta_r \cos e_\theta + \sin \theta_r \sin e_\theta) - \omega e_x \\
 &= \sin e_\theta (\dot{x}_r \cos \theta_r + \dot{y}_r \sin \theta_r) + \cos e_\theta (\dot{y}_r \cos \theta_r - \dot{x}_r \sin \theta_r) - \omega e_x \\
 &= v_r \sin e_\theta - \omega e_x \\
 \dot{e}_\theta &= \frac{d}{dt} (\theta_r - \theta) = \dot{\theta}_r - \dot{\theta} = \omega_r - \omega_c
 \end{aligned}$$

dimana e_x , e_y dan e_θ menyatakan *error leading*, *error lateral*, dan *error orientasi*.

Untuk koreksi *error posture* menggunakan pengukuran MFD yang diolah dengan Fuzzy Mamdani akan dijelaskan pada sub-bab perancangan kontroler. Didefinisikan \bar{e} sebagai hasil koreksi *error tracking*, e sebagai *error* posisi dan orientasi dari pengukuran encoder, dan \hat{e} sebagai *error* posisi dan orientasi dari pengukuran MFD yang memenuhi:

$$\bar{e} = GEe - GM\hat{e} \quad (3.4)$$

dimana GE dan GM adalah *gain* dari DOB.

Dengan menggunakan (3.4), model *error* kinematika (2.14) dapat ditulis sebagai berikut [11]:

$$\begin{bmatrix} \dot{\bar{e}}_x \\ \dot{\bar{e}}_y \\ \dot{\bar{e}}_\theta \end{bmatrix} = \begin{bmatrix} \cos \bar{e}_\theta & 0 \\ \sin \bar{e}_\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} v_r \\ \omega_r \end{pmatrix} + \begin{bmatrix} -1 & \bar{e}_y \\ 0 & -\bar{e}_x \\ 0 & -1 \end{bmatrix} u \quad (3.5)$$

Kontrol *input axiliary velocity* yang digunakan yaitu [4]:

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos \bar{e}_\theta + k_x \bar{e}_x \\ \omega_r + k_y v_r \bar{e}_y + k_\theta v_r \sin \bar{e}_\theta \end{bmatrix} \quad (3.6)$$

Berdasarkan (3.5) dan (3.6) didapatkan *error posture* sebagai berikut [6]:

$$\begin{aligned} \dot{\bar{e}}_x &= -k_x \bar{e}_x + \bar{e}_y \omega_r + k_y \bar{e}_y v_r \bar{e}_y + k_\theta \bar{e}_y v_r \sin \bar{e}_\theta \\ \dot{\bar{e}}_y &= v_r \sin \bar{e}_\theta - \bar{e}_x \omega_r - k_y v_r \bar{e}_y - k_\theta v_r \sin \bar{e}_\theta \\ \dot{\bar{e}}_\theta &= -k_y v_r \bar{e}_y - k_\theta v_r \sin \bar{e}_\theta \end{aligned} \quad (3.7)$$

Dari (3.7) didapatkan model *error* kinematika sebagai berikut [6]:

$$\begin{aligned} \begin{bmatrix} \dot{\bar{e}}_x \\ \dot{\bar{e}}_y \\ \dot{\bar{e}}_\theta \end{bmatrix} &= \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \sin \bar{e}_\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{e}_x \\ \bar{e}_y \\ \bar{e}_\theta \end{bmatrix} \\ &+ \begin{bmatrix} -\bar{e}_x & v_r \bar{e}_y^2 & v_r \bar{e}_y \sin \bar{e}_\theta \\ 0 & -v_r \bar{e}_x \bar{e}_y & -v_r \bar{e}_x \sin \bar{e}_\theta \\ 0 & -v_r \bar{e}_y & -v_r \sin \bar{e}_\theta \end{bmatrix} \begin{bmatrix} k_x \\ k_y \\ k_\theta \end{bmatrix} \end{aligned} \quad (3.8)$$

dimana k_x, k_y , dan k_θ merupakan parameter yang akan ditentukan.

Pada [11] disebutkan bahwa penggunaan 4 *state* bertujuan untuk memisahkan bagian sistem yang *uncontrollable* pada model *error* kinematika 3-*state* sehingga *posture* terbatas pada rentang orientasi 2π . Keterbatasan *posture* ini menyebabkan pemberian *input* pada dua inisial kondisi sama namun θ_0 berbeda 2π akan menghasilkan respon orientasi berbeda walaupun posisi bernilai sama [14].

Keterbatasan orientasi ini yang menyebabkan adanya bagian *uncontrollable* pada model *error* kinematika 3 *state* (2.14) [11] walaupun nilai *state error* orientasi (e_θ) telah dibatasi pada $\pm 90^\circ$. Oleh karena itu akan dilakukan pengecekan *controllability* dari sistem (3.2) terlebih dahulu dengan menggunakan teorema Chow. Teorema ini menyatakan bahwa sebuah sistem adalah *controllable* jika *involutive distribution* berdasarkan Lie Bracket dari sistem memiliki *rank* sama dengan dimensi konfigurasi *space* [14]. Pengecekan ini digunakan untuk nonlinear sistem namun juga dapat dilakukan untuk sistem linear dengan persamaan sebagai berikut:

$$\text{rank}[B, [AB], [A^2B], \dots] = n \quad (3.9)$$

Sedangkan Lie Bracket dapat digunakan untuk menentukan arah pergerakan dari *differential drive* [14]. Misal f dan g adalah dua bidang vektor, Lie Bracket dari f dan g merupakan sebuah bidang vektor ketiga [14]:

$$[f, g] = \nabla g f - \nabla f g \quad (3.10)$$

Berdasarkan nonholonomic *constraints*, AGV dengan model kinematika (3.2) tidak dapat bergerak ke arah *lateral* dan hanya dapat bergerak sesuai dengan orientasi roda. Sehingga vektor dari pergerakan AGV dapat didefinisikan oleh $s_1(p)$ dan $s_2(p)$ [14]:

$$s_1(q) = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}, \quad s_2(q) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.11)$$

Menggunakan (3.9) dan (3.10) didapatkan arah pergerakan baru [14]:

$$s_3 = [s_1, s_2] = \frac{\partial s_2}{\partial p} s_1 - \frac{\partial s_1}{\partial p} s_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} - \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s_3 = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \quad (3.12)$$

Dengan menghitung Lie Bracket dari kombinasi s_1 , s_2 , dan s_3 tidak akan dihasilkan *linearly independent* arah pergerakan baru. Kemudian untuk *rank* yang dihasilkan adalah sebagai berikut [15]:

$$\text{rank}[s_1, s_2, s_3] = \text{rank} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ \sin \theta & 0 & -\cos \theta \\ 0 & 1 & 0 \end{bmatrix} = 3 \quad (3.13)$$

Sistem (3.2) memiliki 3 state dan *rank* berdasarkan Lie Bracket sebesar 3 sehingga menggunakan teorema chow [14] dapat dinyatakan bahwa sistem adalah *controllable*. Kemudian dilakukan pengecekan *controllability* dari sistem (2.14) dan (3.8) dengan menggunakan (3.9) dan bantuan MATLAB-script seperti pada Lampiran B.1 dan B.2. Untuk sistem (2.14), bentuk dari input kontrol (u) adalah sebagai berikut:

$$u = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_r \cos \bar{e}_\theta + v_b \\ \omega_r + \omega_b \end{bmatrix} \quad (3.14)$$

dimana v_b dan ω_b merupakan sinyal *feedback*.

Ketika nilai v_r dan ω_r bernilai konstan, dapat diketahui dengan mudah matriks *controllability* untuk (2.14) dengan *input* kontrol (3.14) adalah *full rank*. Namun untuk nilai v_r dan ω_r yang berubah-ubah, penentuan *rank* dari matriks *controllability* menjadi lebih rumit [14]. Dari hasil pengecekan menggunakan program pada lampiran C.1 dan C.2 diketahui matrik *controllability* sistem (2.14) dan (3.8) adalah *full rank* pada semua kondisi kecuali saat ω_r dan v_r bernilai nol. Sehingga dengan perancangan v_r dan ω_r yang tepat kondisi *uncontrollable* pada model *error* kinematika dapat dihindari.

3.1.2 Struktur dan Model Dinamika

Persamaan dinamika diperoleh dari pemodelan menggunakan *lagrange dynamic approach* [8]. Dengan mengacu pada pembahasan sub-bab sebelumnya, model dari dinamika sistem dinyatakan oleh persamaan berikut:

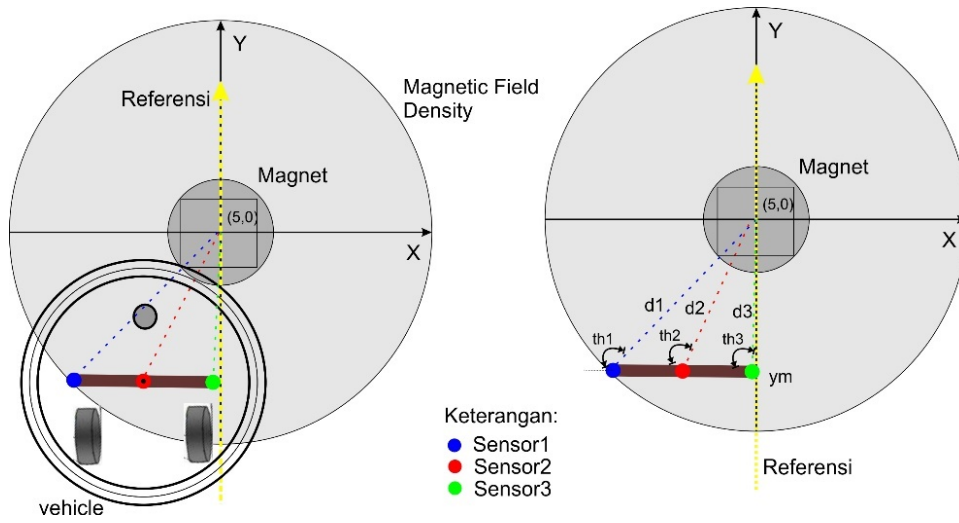
$$\bar{M}(q)\dot{\eta} + \bar{V}(q, \dot{q})\eta = \bar{B}(q)\tau \quad (3.15)$$

dimana $\tau = [\tau_R, \tau_L]^T$ merupakan *input* kontrol torsi dan $\eta = [\omega_R, \omega_L]^T$ merupakan kecepatan angular roda kanan dan kiri.

Parameter $\bar{M}(q)$, $\bar{V}(q, \dot{q})$, dan $\bar{B}(q)$ adalah sebagai berikut [8]:

$$\bar{M}(q) = \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 + I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix} \quad (3.16)$$

$$\bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & I_w + \frac{R^2}{4L^2}m_c d\dot{\theta} \\ I_w + \frac{R^2}{4L^2}m_c d\dot{\theta} & 0 \end{bmatrix}, \bar{B}(q) = \frac{1}{r} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



Gambar 3.4 Perancangan Sensor *Magnetic* dan Pembacaan *Flux* dengan

$$\begin{aligned}\bar{M}(q) &= S(q)^T(q)M(q)S(q) \\ \bar{V}(q, \dot{q}) &= S(q)^T(q)M(q)\dot{S}(q) + S^T V(q, \dot{q})S(q) \\ \bar{B}(q) &= S(q)^T(q)B(q)\end{aligned}$$

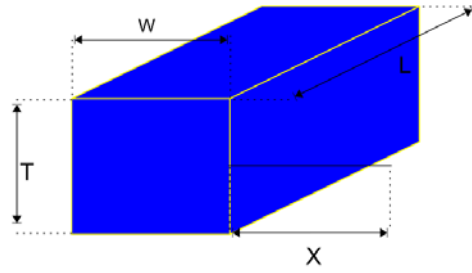
Model dinamika (3.16) dapat disederhanakan menggunakan model kinematika sebagai berikut:

$$\begin{aligned}\left(m + \frac{2I_w}{R^2}\right)\dot{v} - m_c d \omega^2 &= \frac{1}{R}(\tau_R + \tau_L) \\ \left(I + \frac{2L^2}{R^2}I_w\right)\dot{\omega} + m_c d \omega v &= \frac{L}{R}(\tau_R - \tau_L)\end{aligned}\tag{3.17}$$

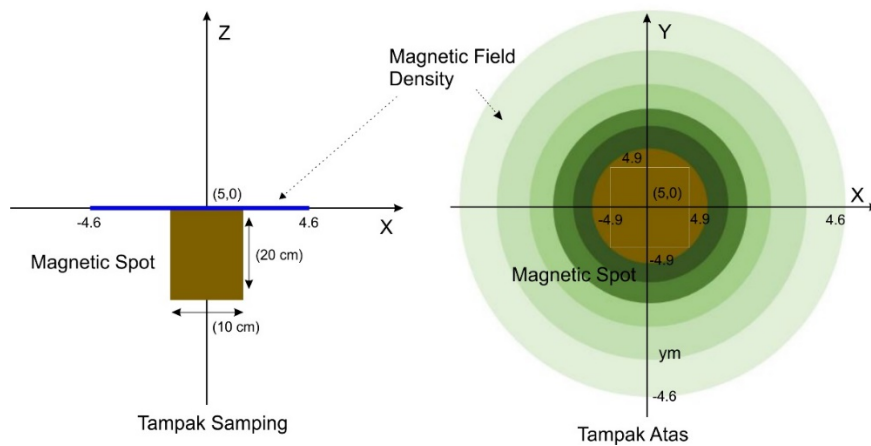
dimana d adalah jarak antara titik pusat AGV dengan roda, I_w adalah momen inersia roda dengan motor pada titik sumbu roda.

3.2 Pemodelan Magnet dan Pembacaan Sensor

Untuk koreksi *error* pergerakan dari AGV berdasarkan *feedback* dari pengukuran encoder digunakan *magnetic spot* sebagai referensi eksternal dengan memanfaatkan perbedaan nilai yang terukur pada area MFD menggunakan sensor hall-effect. Pada Gambar 3.4 ditunjukkan ilustrasi pembacaan sensor magnet pada area MFD menggunakan 3 sensor. MFD dirancang menggunakan *finite element method* dengan bantuan software FEMM.



Gambar 3.5 Permanent Magnet



Gambar 3.6 Rancangan Magnetic Spot

Persamaan dari MFD dapat dinyatakan sebagai berikut:

$$B_X = \frac{B_r}{\pi} \left(\tan^{-1} \frac{LT}{2X\sqrt{4X^2 + L^2 + T^2}} - \tan^{-1} \frac{LT}{2(W+X)\sqrt{4(T+X)^2 + L^2 + T^2}} \right) \quad (3.18)$$

dimana

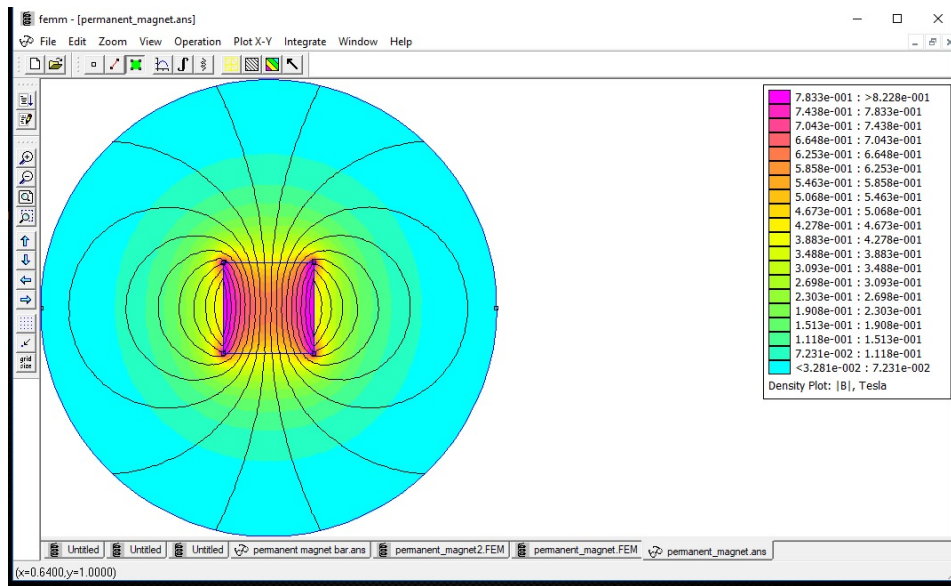
B_r : residual flux density

B_X : magnetic flux density (MFD)

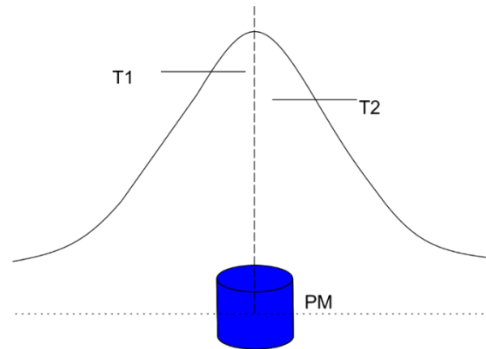
X : surface magnetic flux of permanent magnet

(L, T, W) : panjang, ketebalan, dan lebar permanent magnet

Perhitungan untuk nilai X ditunjukkan pada Lampiran C.3. Parameter yang digunakan sesuai dengan datasheet pada Intemag untuk *permanent magnet* dengan material NdFeb sedangkan untuk residual flux density (B_r) bernilai 13.300 gauss.



Gambar 3.7 Distribusi MFD dengan FEMM

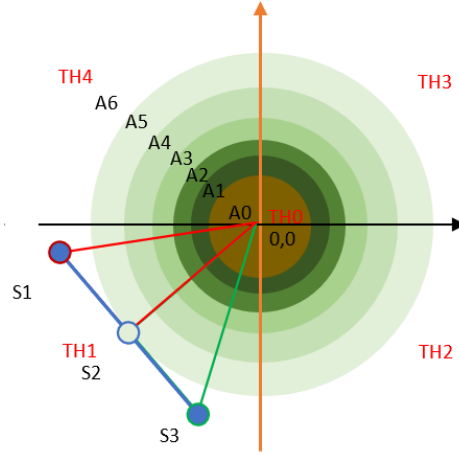


Gambar 3.8 *Threshold* untuk Pembacaan MFD

Nilai *permeability* dari *permanent magnet* adalah 1.05 dan *coercivity* dihitung dengan persamaan berikut:

$$H_c = 155319 * \sqrt{\frac{BH_{max}}{MGoe}} \quad (3.19)$$

dimana BH_{max} adalah *Maximum Energy product* (energy density=42MGoe), MGoe adalah *auxiliary magnetic field* (energy density= 10^6 gauss) dari H , dan H_c merupakan *coercivity* (A/m). Kemudian untuk generator dari MFD digunakan software FEMM dengan parameter yang disesuaikan dengan jenis material NdFeb. Untuk membedakan koordinat positif dan negatif dari sumbu Y digunakan nilai *threshold* sebagai pendeteksi perubahan tanda seperti yang ditunjukkan pada Gambar 3.9.



Gambar 3.9 Pembagian Area MFD

3.3 Perancangan Kontroler

Kontroler dirancang untuk meminimumkan *error tracking* agar posisi dan orientasi AGV tetap berada pada *trajectory* referensi. Perancangan kontroler meliputi (1) perancangan fuzzy Mamdani untuk koreksi *input error* berdasarkan perhitungan DOB, (2) perancangan kontroler fuzzy Takagi-Sugeno dengan skema PDC dan perhitungan LMI, dan (3) perancangan kontoller dinamika menggunakan *dynamic nonlinear feedback*.

3.3.1 Perancangan Koreksi *Input Model Error* Kinematika

Dari Gambar 3.1 dapat dilihat *input* kontroler yang dirancang pada tesis ini terdiri dari dua pengukuran yaitu pengukuran encoder dan sensor magnet. Untuk *error* yang dihasilkan dari pengukuran encoder dapat dilihat pada (3.4). Untuk memperbaiki *error tracking* digunakan sensor magnet dengan bentuk *feedback* sebagai berikut:

$$\begin{aligned} \hat{e} &= P_r - P_m \\ \begin{bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_\theta \end{bmatrix} &= \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix} \end{aligned} \quad (3.20)$$

dimana P_m merupakan hasil pembacaan koordinat posisi dan orientasi AGV pada area MFD dan \hat{e} adalah *error tracking* dari pengukuran sensor magnet. Kedua *error* dari pengukuran encoder dan sensor magnet digunakan sebagai *input model error* kinematika 3 *state*.

Permasalahan yang muncul dari penggunaan dua sensor yaitu berapa besar pembagian nilai *feedback* karena masing-masing pengukuran memiliki kelebihan dan kekurangan. Kelebihan dari encoder yaitu dapat digunakan untuk pengukuran kontinu sehingga perhitungan posisi dan orientasi dapat dilakukan. Sedangkan untuk sensor magnet memiliki kelebihan untuk pengukuran jarak pendek yaitu pada area MFD. Namun sensor magnet tidak dapat digunakan untuk menentukan koordinat posisi AGV berada tanpa referensi dari sensor lain.

Untuk mengatasi hal tersebut pada tesis ini diusulkan penggunaan DOB untuk pemilihan *gain* dari kedua pengukuran. Dengan demikian koreksi dari *input* model *error* kinematika terdiri dari dua langkah yaitu penggunaan DOB untuk menentukan besaran *gain* dari masing-masing pengukuran dan penggunaan fuzzy Mamdani untuk inferensi *output*.

3.3.1.1 Perancangan *Gain Error* Berdasarkan *Degree of Belief*

Konsep dari DOB dibangun berdasarkan letak AGV pada *trajectory* referensi dan area MFD. Letak pada *trajectory* referensi yang dimaksud adalah AGV berada didalam atau diluar area MFD. Berdasarkan konsep tersebut area MFD dibagi menjadi 7 dimana A6 menyatakan area diluar MFD, A0 menyatakan area titik maksimum MFD, dan TH menyatakan kuadran yang ditentukan dari arah *trajectory* referensi.

Namun area pada Gambar 3.9 dapat berubah sesuai dengan *trajectory* yang diinginkan. Dengan B adalah nilai MFD, GE adalah *gain error* dari sensor encoder, dan GM adalah *gain error* dari sensor magnet, pengaturan DOB untuk pemilihan *gain* adalah sebagai berikut:

1. Ketika $B \neq 0$

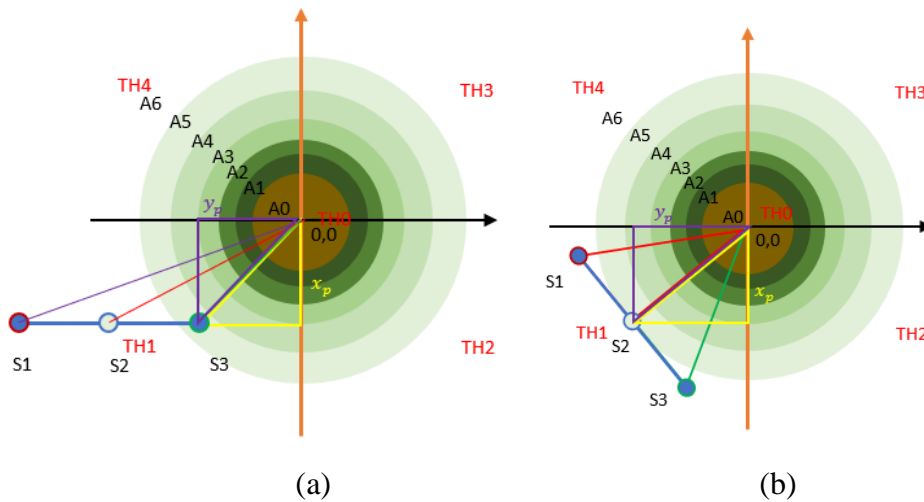
$$GM = \gamma \left(1 - \frac{a}{r}\right), \quad a \leq r \quad (3.21)$$

$$GE = 1 - GM \quad (3.22)$$

dimana a adalah pengukuran MFD yang dirubah menjadi jarak dari titik maksimum, r merupakan referensi jari-jari MFD, dan γ merupakan koefisien referensi.

2. Ketika $B = 0$

Nilai *gain* GM=0 dan GE=1

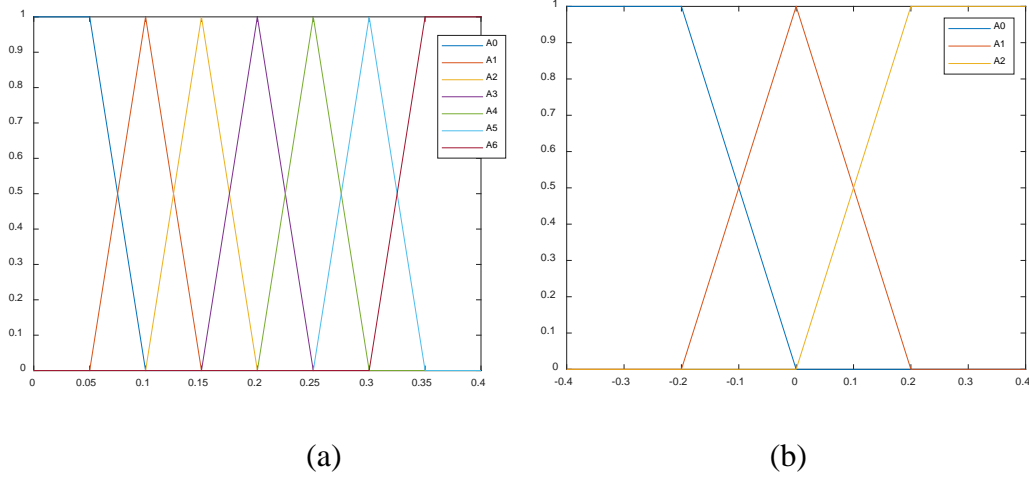


Gambar 3.10 Pembacaan Sensor Magnet (a) Saat S3 membaca (b) Saat S2 Membaca

3.3.1.2 Perancangan Koreksi *Error Tracking* dari Pengukuran Sensor Magnet

Untuk mengolah pengukuran MFD digunakan fuzzy Mamdani. Pengolahan ini dibutuhkan untuk mempermudah perhitungan posisi dan orientasi AGV dari hasil pengukuran MFD dan meminimalkan kesalahan pengukuran akibat adanya gangguan eksternal. Dengan menggunakan PM berdimensi 10x10x20 cm (maksimum yang terdapat dipasaran) hasil distribusi dari MFD ditunjukkan pada Gambar 3.8. Karena FEA merupakan metode numerical dan tidak terdapat analisis vektor, hasil pengukuran tidak bisa digunakan untuk menentukan koordinat posisi dan orientasi AGV secara langsung sehingga digunakan tiga sensor magnet 1-axis seperti yang terlihat pada Gambar 3.4.

Variabel *input* yang digunakan adalah pengukuran sensor tengah, selisih sensor kanan dan sensor kiri, dan *threshold*. Variabel output yang dirancang yaitu x_{mp} , y_{mp} , dan θ_{mp} . Membership function dari *input* sensor 2 diatur sesuai dengan distribusi MFD dengan deskripsi linguistik: A6; A5; A4; A3; A2; A1. Kemudian untuk *input* kedua yaitu selisih antara sensor kanan dan sensor kiri sebagai referensi penentuan orientasi AGV dari koordinat titik maksimum MFD dengan deskripsi linguistik: A0; A1; A3. Untuk A0 pada Gambar 3.11b menunjukkan sensor kiri lebih dekat dengan titik maksimum MFD. Untuk *input* ketiga yaitu *threshold* digunakan untuk menentukan letak kuadran AGV pada area MFD dengan deskripsi linguistik: A1; A2; A3; A4.



Gambar 3.11 Membership function (a) Sensor 2 (b) Selisih Sensor 1 dan 3

Untuk defuzzifikasi digunakan *center of gravity* dengan persamaan sebagai berikut:

$$u^{tegas} = \frac{\sum_i b_i \int \mu_i}{\sum_i \int \mu_i} \quad (3.23)$$

dimana b_i adalah titik tengah fungsi keanggotaan dan $\int \mu_i$ adalah area dibawah fungsi keanggotaan μ_i .

3.3.2 Perancangan Kontroler Fuzzy Takagi-Sugeno

Perancangan kontroler pada tesis ini dilakukan dengan menggiring permasalahan dinamika menjadi permasalahan kinematika dengan menggunakan (3.6) dimana k_x, k_y , dan k_θ merupakan parameter yang akan dihitung dengan PDC.

Salah satu pendekatan dalam penentuan aturan *if premise then consequences* dari kontroler fuzzy Takagi-Sugeno yaitu dengan melakukan linearisasi pada titik-titik kerja untuk menghasilkan model linear. Sedangkan aturan *plant* untuk tiap model linier adalah sebagai berikut:

$$\begin{aligned} &\text{if } z_1(t) \text{ is } M_{i1} \text{ and } z_2(t) \text{ is } M_{i2} \dots z_p \text{ is } M_{ip} \\ &\text{then } \dot{e}(t) = A_i e(t) + B_i q(t) \quad ; i = 1, 2, \dots, r \\ &y(t) = C_i e(t) \end{aligned} \quad (3.24)$$

dimana r adalah jumlah aturan, M_{ij} adalah fungsi keanggotaan fuzzy, $e(t)$ adalah *state* vektor, $q(t)$ adalah *input* vektor, dan $z_1(t), \dots, z_p(t)$ merupakan variabel *premise* yang diketahui.

Dengan menggunakan *state* vektor ($e(t)$) dan *input* vektor ($q(t)$), fuzzy T-S model dapat dibentuk sebagai berikut:

$$\dot{\bar{e}}(t) = \sum_{i=1}^r h_i(z(t)) \{A_i \bar{e}(t) + B_i q(t)\} \quad (3.25)$$

$$y(t) = \sum_{i=1}^r h_i(z(t)) C_i \bar{e}(t) \quad (3.26)$$

dengan

$$h_i(t) = \frac{w_i(t)}{\sum_{i=1}^r w_i(z(t))}$$

Pada tesis ini digunakan model *error tracking* kinematika 3 *state* sesuai (3.8) dan fuzzy T-S model (3.25) dapat ditulis menjadi:

$$\dot{\bar{e}}(t) = A_z \bar{e}(t) + B_z q(t) \quad (3.27)$$

dengan

$$A_z = \sum_{i=1}^r h_i(z(t)) A_i, \quad B_z = \sum_{i=1}^r h_i(z(t)) B_i,$$

Berdasarkan model sistem (3.8) matrik A_i dan B_i yaitu

$$A_i = \begin{bmatrix} 0 & z_1 & 0 \\ -z_1 & 0 & z_2 z_5 \\ 0 & 0 & 0 \end{bmatrix}, B_i = \begin{bmatrix} -z_3 & z_6 & z_2 z_4 z_5 \\ 0 & -z_2 z_3 z_4 & -z_2 z_3 z_5 \\ 0 & -z_2 z_4 & -z_2 z_5 \end{bmatrix} \quad (3.28)$$

dimana jumlah aturan fuzzy adalah $2^6 = 64$.

Untuk stabilisasi fuzzy T-S model pada (3.27) digunakan PDC seperti pada pembahasan Bab 2 dengan kontrol *law*:

$$q(t) = - \sum_{i=1}^r h_i(z(t)) K_i \bar{e}(t), i = 1, 2, 3, \dots, r \quad (3.29)$$

dimana $q = [v, \omega]^T$ dan K_i merupakan *gain* yang akan dihitung menggunakan LMI.

Sistem lup terbuka dari (3.25) adalah:

$$\dot{\bar{e}}(t) = \sum_{i=1}^r h_i(z(t)) A_i \bar{e}(t) \quad (3.30)$$

Sistem (3.30) adalah stabil asimtotik global jika terdapat matriks P yang memenuhi (2.44). Jumlah aturan (r) yang digunakan adalah 64 dan P merupakan matriks stabilitas yang dicari. Kondisi kestabilan dari sistem (3.30) akan dicari dengan memilih kandidat fungsi Lyapunov sebagai berikut:

$$V(\bar{e}(t)) = \bar{e}(t)^T P \bar{e}(t) \quad (3.31)$$

Jika terdapat matriks positif definit P dan $V(e(t))$ pada semua t , kestabilan sistem (3.30) dapat dibuktikan dengan (3.31).

Substitusi (3.29) ke (3.25) diperoleh sistem lup tertutup sebagai berikut:

$$\dot{\bar{e}}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) \{A_i - B_i K_j\} \bar{e}(t) \quad (3.32)$$

Dari (3.32) didefinisikan:

$$G_{ij} = A_i - B_i K_j \quad (3.33)$$

dengan menggunakan (3.33) pada sistem (3.32), diperoleh persamaan berikut [12]:

$$\begin{aligned} \dot{\bar{e}} = & \sum_{i=1}^r h_i(z(t)) h_j(z(t)) G_{ii} \bar{e}(t) \\ & + 2 \sum_{i=1}^r \sum_{i < j} h_i(z(t)) h_j(z(t)) \left\{ \frac{G_{ij} + G_{ji}}{2} \right\} \bar{e}(t) \end{aligned} \quad (3.34)$$

dimana $i < j$ dan $h_i \cap h_j \neq \emptyset$.

Berdasarkan kondisi kestabilan dari sistem lup terbuka (2.45), kondisi kestabilan dari sistem lup tertutup (3.32) dapat ditentukan menggunakan pertidaksamaan berikut ini [12]:

$$G_{ii}^T P + P G_{ii} < 0 \quad (3.35)$$

$$\left(\frac{G_{ij} + G_{ji}}{2} \right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2} \right) \leq 0 \quad (3.36)$$

Sesuai dengan pembahasan pada sub-bab 2.2 jika terdapat matriks positif definit P memenuhi (3.35) dan (3.36), kondisi kestabilan sistem (3.32) adalah stabil asimtotik global [12].

Substitusi (3.33) ke (3.35) diperoleh:

$$(A_i - B_i K_i)^T P + P (A_i - B_i K_i) < 0, \quad i = 1, 2, 3, \dots, r \quad (3.37)$$

Diketahui $PP^{-1} = I$ sehingga (3.37) dapat diubah seperti berikut:

$$\begin{aligned}
P^{-1}((A_i - B_i K_i)^T P + P(A_i - B_i K_i)) &< 0 \\
P^{-1}(A_i - B_i K_i)^T P + P^{-1}P(A_i - B_i K_i) &< 0 \\
P^{-1}((A_i - B_i K_i)^T P + (A_i - B_i K_i)) &< 0 \\
P^{-1}((A_i - B_i K_i)^T P P^{-1} + (A_i - B_i K_i)) P^{-1} &< 0 \\
P^{-1}A_i^T - P^{-1}B_i^T K_i^T + A_i P^{-1} - B_i K_i P^{-1} &< 0
\end{aligned} \tag{3.38}$$

Didefinisikan variabel $X = P^{-1}$ dan masukkan ke (3.38):

$$XA_i^T - XB_i^T K_i^T + A_i X - B_i K_i X < 0 \tag{3.39}$$

Misal $M_i = K_i X$ dan masukkan ke (3.39) sehingga didapatkan:

$$XA_i^T - M_i^T B_i^T + A_i X - B_i M_i < 0 \tag{3.40}$$

Dengan prosedur yang sama, bentuk dari (3.36) dapat diubah menjadi:

$$XA_i^T - M_i^T B_i^T + A_i X - B_i M_i + XA_j^T - M_j^T B_j^T + A_j X - B_j M_j \leq 0 \tag{3.41}$$

dengan demikian jika terdapat matriks positif definit X dan M yang memenuhi (3.40) dan (3.41) maka sistem pada (3.32) adalah *globally asymptotically stable* [13] dimana *gain* dari fuzzy kontroler yang dicari yaitu $K_i = M_i P$ dan matriks stabilitas didapatkan dari $P = X^{-1}$.

Dalam tesis ini dirancang *constraint input* pada (3.42) dan (3.43). Diasumsikan inisial kondisi $\bar{e}(0)$ diketahui sehingga *constraint* pada *input* yaitu $\|v_d\|_2 \leq \mu$ akan berlaku pada semua waktu $t \geq 0$ jika LMI memenuhi:

$$\begin{bmatrix} 1 & \bar{e}(0)^T \\ \bar{e}(0) & X \end{bmatrix} \geq 0 \tag{3.42}$$

$$\begin{bmatrix} X & M_i^T \\ M_i & \mu^2 I \end{bmatrix} \geq 0 \tag{3.43}$$

dengan

$$X = P^{-1} \text{ dan } M_i = K_i X$$

Diasumsikan (3.31) adalah fungsi Lyapunov dan nilai $\bar{e}(0)^T P \bar{e}(0) \leq 1$ sehingga:

$$1 - \bar{e}(0)^T X^{-1} \bar{e}(0) \geq 0 \tag{3.44}$$

Dengan menggunakan *Schur complement*, (3.44) ditransformasi menjadi (3.42).

Pada tesis ini digunakan *constraint output* dalam bentuk pertidaksamaan berikut:

$$\begin{bmatrix} X & XC_i^T \\ C_i X & \lambda^2 I \end{bmatrix} \geq 0 \tag{3.45}$$

Kecepatan respon dipengaruhi oleh nilai *decay rate* [5]. Respon ini dapat ditentukan dari nilai terbesar ekponensial fungsi Lyapunov. Kondisi dari (3.40) dan (3.41) yang memenuhi $\dot{V}(e(t)) \leq -2\alpha V(e(t))$ pada semua *trajectory* yaitu [13]:

$$G_{ii}^T P + P G_{ii} + 2\alpha P < 0, \quad \alpha > 0 \quad (3.46)$$

$$\left(\frac{G_{ij} + G_{ji}}{2}\right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2}\right) + 2\alpha P \leq 0, \quad \alpha > 0 \quad (3.47)$$

dimana α merupakan *decay rate*.

Berdasarkan (3.46) dan (3.47) bentuk dari (3.40) dan (3.41) dapat ditulis menjadi sebagai berikut:

$$X A_i^T - M_i^T B_i^T + A_i X - B_i M + 2\alpha X < 0 \quad (3.48)$$

$$\begin{aligned} X A_i^T - M_j^T B_i^T + A_i X - B_i M_j + X A_j^T \\ - M_i^T B_j^T + A_j X - B_j M_i - 4\alpha X \leq 0 \end{aligned} \quad (3.49)$$

Dengan demikian permasalahan kontrol pada tesis ini yaitu bagaimana menemukan matriks stabilitas P yang memenuhi LMI (2.45), (3.42), (3.43), (3.45), (3.48), dan (3.49) untuk perhitungan matriks *gain* $K_i = M_i P$ dan parameter k_x , k_y , dan k_θ dari *input auxiliary velocity*. Untuk MATLAB-Script dari permasalahan LMI ini dapat dilihat pada lampiran C.5.

3.3.3 Perancangan Kontroler Dinamika

Dari perancangan kontroler fuzzy T-S didapatkan *input* kontrol *auxiliary velocity* (3.6) dalam bentuk kecepatan linier dan angular AGV. Kecepatan ini kemudian diubah kedalam bentuk kecepatan roda kanan dan kiri menggunakan (3.1) dan didefinisikan *dynamic error tracking* dari AGV sebagai berikut:

$$\tilde{e} = \eta - q \quad (3.50)$$

dimana \tilde{e} merupakan *dynamic error tracking*, q merupakan *input auxiliary velocity*, dan η merupakan kecepatan angular roda kanan dan kiri AGV.

Turunan dari (3.6) terhadap waktu sebagai berikut [7]:

$$\dot{q} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{v}_r \cos e_\theta \\ \dot{\omega}_r + k_y \dot{v}_r e_y \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r \sin e_\theta \\ 0 & k_2 v_r & k_\theta v_r \cos e_\theta \end{bmatrix} \quad (3.51)$$

dengan *input* kontrol yang dirancang adalah sebagai berikut:

$$u = K_r \dot{\eta} + K_p \tilde{e} (1 + T_d) \quad (3.52)$$

dimana K_r merupakan matrik diagonal sebagai *forward gain* untuk *auxiliary velocity*, u merupakan *input* perhitungan kontrol torsi, K_p merupakan matrik diagonal dengan nilai besar sebagai *gain proportional*, dan $K_p T_d$ merupakan matrik diagonal yang bernilai kecil sebagai *gain derivative*.

dengan mengatur *forward gain* untuk *auxiliary velocity* dan menggunakan kontroler *proportional* dan *derivative* untuk *gain error* dari *dynamic nonlinear feedback* didapatkan perhitungan torsi untuk dinamika sebagai berikut:

$$\tau = \bar{B}^{-1}(q)(\bar{M}(q)[K_r \dot{q} + K_p \bar{e}(1 + T_d)] + \bar{V}(q, \dot{q})\eta) \quad (3.53)$$

3.3.4 Analisa Steady State Error

Untuk analisa *steady state error* digunakan (3.6) sebagai *input auxiliary velocity* dan dibuktikan bahwa $p_e = 0$ adalah sebuah titik *uniformly asymptotically stable* ketika $t \rightarrow \infty$ dalam *tracking* dengan kondisi $v_r > 0$ untuk semua t .

Pertama dilakukan linearisasi persamaan differensial (3.7) disekitar titik $P_e = 0$ sebagai berikut [14]:

$$\dot{P}_e = A(t)P_e \quad (3.54)$$

dengan

$$A(t) = \left. \frac{\partial f(t, P_e)}{\partial P_e} \right|_{p_e=0} = \begin{bmatrix} -K_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -v_r K_y & -v_r K_\theta \end{bmatrix}$$

Karakteristik persamaan dari $A(t)$ adalah sebagai berikut [17]:

$$a_3 s^3 + a_2 s^2 + a_1 s + a_0 = 0 \quad (3.55)$$

Untuk mencari parameter a dari (3.55) dapat digunakan:

$$s^3 - \text{tr}(A)s^2 + (\text{tr}(A)^2 - \text{tr}(A^2))s + \det(A) \quad (3.56)$$

Dengan menggunakan (3.56) parameter dari karakteristik polinomial (3.55) dapat ditentukan sebagai berikut:

$$a_3 = 1$$

$$a_2 = \text{tr}(A)$$

$$= K_\theta v_r + K_x$$

$$a_1 = \text{tr}(A)^2 - \text{tr}(A^2)$$

$$= K_x^2 + K_x v_r K_\theta + v_r^2 K_\theta^2 - (K_x^2 - \omega_r^2 - v_r^2 K_y + v_r^2 K_\theta^2)$$

$$= K_y v_r^2 + K_x K_\theta v_r + \omega_r^2$$

$$a_0 = K_x K_y v_r^2 + \omega_r^2 K_\theta v_r$$

Kemudian disusun koefisien *array* sebagai berikut [17]:

$$\begin{array}{ccc} s^3 & a_0 & a_2 \\ s^2 & a_1 & a_3 \\ s^1 & \frac{a_1 a_2 - a_0 a_3}{a_1} & \\ s^0 & a_3 & \end{array}$$

Kondisi agar semua akar bernilai negatif adalah:

$$a_h = a_1 a_2 - a_0 a_3, \quad a_h > 0 \quad (3.57)$$

Nilai a_h dihitung menggunakan parameter dari (3.56) sebagai berikut:

$$\begin{aligned} a_h &= (K_\theta v_r + K_x)(K_y v_r^2 + K_x K_\theta v_r + \omega_r^2) - K_x K_y v_r^2 - \omega_r^2 K_\theta v_r \\ a_h &= K_\theta K_y v_r^3 + K_\theta v_r K_x K_\theta v_r + K_x K_x K_\theta v_r + K_x \omega_r^2 \geq 0 \end{aligned} \quad (3.58)$$

Pada (3.58) terdapat kemungkinan a_h bernilai nol sehingga akan digunakan *relative stability* dari kestabilan Routh dengan cara menggeser sumbu *s-plane* baru kemudian diterapkan kriteria kestabilan [17]. Hal ini dapat dilakukan dengan substitusi $s = z - \sigma$ pada karakteristik polinomial (3.53) dimana σ merupakan sebuah konstanta, sehingga didapatkan:

$$b_3 z^3 + b_2 z^2 + b_1 z + b_0 = 0 \quad (3.59)$$

Kemudian dipilih nilai α, β , dan γ yaitu $K_x/3, K_\theta/3$, dan K_θ/K_y sehingga didapatkan nilai a_h yang memenuhi (3.55) sebagai berikut:

$$\begin{aligned} b_1 b_2 - b_0 b_3 &= \omega_r^2 (\alpha + \sigma) + K_y v_r^2 (\beta + \sigma) + (K_\theta v_r + \alpha)(\alpha \beta + \sigma \alpha + \sigma \beta) \\ &+ (\alpha + 2\sigma)(\beta + 2\sigma)\sigma > 0 \end{aligned}$$

Berdasarkan kriteria Routh-Hurwitz dan nilai q_r yang telah dirancang, semua akar pada (3.57) akan bernilai negatif sehingga semua akar dari karakteristik polinomial (3.53) akan bernilai kurang dari atau sama dengan $-\sigma$. Karena matriks $A(t)$ merupakan Hurwitz sehingga titik $P_e = 0$ merupakan titik *uniformly asymptotically stable* dari (3.52) berdasarkan Lypunov *indirect method* [18].

Selanjutnya akan dilakukan analisa terhadap *steady state error* berdasarkan fuzzy T-S model (3.8) dengan menggunakan fungsi Lyapunov. Apabila A_i merupakan sebuah matriks Hurwitz, akan terdapat matriks positif definit $P = P^T$ yang memenuhi (2.43).

Sesuai dengan perancangan PDC dipilih fungsi Lyapunov dari linearisasi sistem yang terdapat pada (3.8) sebagai berikut [13]:

$$V(t) = \bar{e}(t)^T P \bar{e}(t) \quad (3.60)$$

Turunan dari (3.58) terhadap waktu adalah [19]:

$$\begin{aligned} \dot{V}(t) &= \dot{\bar{e}}(t)^T P \bar{e}(t) + \bar{e}(t)^T P \dot{\bar{e}}(t) = \bar{e}(t)^T (A_i^T P + P A_i) \bar{e}(t) \\ &= -\bar{e}(t)^T Q \bar{e}(t) \end{aligned} \quad (3.61)$$

Dengan demikian berdasarkan Lyapunov *direct method*, fungsi $V(t)$ merupakan sebuah fungsi Lyapunov dan titik $P_e = 0$ adalah stabil global apabila terdapat matriks positif definit $Q = Q^T$ yang memenuhi [12]:

$$A_i^T P + P A_i = -Q \quad (3.62)$$

Titik equilibrium $P_e = 0$ adalah *asymptotically stable* bila terdapat matriks positif definit P pada (3.60) dan $\dot{V}(t)$ adalah negatif definit yang memenuhi [19]:

$$\begin{aligned} V(t) &> 0 \\ \dot{V}(t) &< -e(t)^T Q e(t) < 0 \end{aligned} \quad (3.63)$$

Berdasarkan analisa dari fungsi Lyapunov dan perancangan PDC dengan perhitungan LMI, matriks parameter Q pada (3.60) dapat dihitung menggunakan matriks stabilitas P yang memenuhi pertidaksamaan (3.42), (3.43), (3.45), (3.48), dan (3.49) .

Selanjutnya dilakukan analisis untuk kontroler dinamika berdasarkan kontrol *input* yang digunakan (3.52). Didefinisikan *input auxiliary velocity error* dari (3.50) sebagai berikut [7]:

$$\tilde{e} = \eta - q = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \begin{bmatrix} v_1 - v_r \cos e_\theta - k_x e_x \\ v_2 - w_r - k_y v_r e_y - k_3 v_r \sin e_\theta \end{bmatrix} \quad (3.64)$$

dimana η merupakan kecepatan linier roda kanan dan kiri dan q merupakan *input auxiliary velocity*.

Dengan menggunakan (3.52) didapatkan [7] [15]:

$$\dot{\tilde{e}} = -K_p \tilde{e} \quad (3.65)$$

Berdasarkan perancangan kontroler fuzzy T-S dan asumsi pada kontroler dinamika, vektor *auxiliary velocity* akan konvergen secara eksponensial menuju nol yang memenuhi $\eta \rightarrow q$ seiring $t \rightarrow \infty$.

Kemudian menggunakan (3.60), (3.64) dan (3.65) dipilih kandidat fungsi Lyapunov sebagai berikut:

$$V(t) = \bar{e}(t)^T P \bar{e}(t) + \frac{1}{2K_p} \left(e_4^2 + \frac{k_x}{k_y k_\theta v_r} e_5^2 \right) \quad (3.66)$$

Berdasarkan perancangan PDC dapat dipastikan bahwa $V(t) \geq 0$ kemudian nilai $V = 0$ hanya terjadi ketika $\bar{e} = 0$ dan $\tilde{e} = 0$. Untuk turunan dari (3.66) terhadap waktu adalah:

$$\dot{V}(t) = -\bar{e}(t)^T Q \bar{e}(t) - (e_4 + k_x e_x)^2 - \frac{k_x}{k_y k_\theta v_r} (e_5 + k_\theta v_r \sin e_\theta)^2 \quad (3.67)$$

Dengan demikian akan didapatkan $\dot{V} \leq 0$ kemudian karena nilai $V(t)$ tidak meningkat dan menuju nilai tertentu berdasarkan Barbalat lemma akan didapatkan $\dot{V} \rightarrow 0$ seiring $t \rightarrow \infty$ [14]. Dengan menimbang bahwa $\tilde{e} \rightarrow 0$ seiring $t \rightarrow \infty$ didapatkan limit dari (3.67) sebagai berikut:

$$0 = -[e_x \quad e_y \quad e_\theta] Q \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \quad (3.68)$$

Dari (3.68) dapat terlihat bahwa $[e_x \quad e_y \quad e_\theta]^T \rightarrow 0$ seiring $t \rightarrow \infty$, sehingga dapat disimpulkan bahwa titik *equilibrium* $e = 0$ merupakan sebuah titik *uniformly asymptotically stable* dengan syarat didapatkan matriks positif definit Q yang memenuhi (3.61) dan (3.63).

3.3.5 Rencana Pengujian

Beberapa rencana pengujian pada tesis ini adalah sebagai berikut:

1. Pengujian kontroler fuzzy T-S menggunakan model kinematika untuk *trajectory tracking*.
 - Hasil yang akan diukur yaitu kecepatan linier, kecepatan angular dan *error tracking*.
 - Kriteria pengujian ini yaitu untuk kecepatan linier dan sudut *settling time* konvergensi *error tracking* dapat tercapai dalam $t < 15s$ Sedangkan untuk *error tracking* nilai *settling time* dapat tercapai dalam $t < 10s$.
2. Pengujian perbandingan kontroler menggunakan model kinematika 3 *state* dan 4 *state* untuk *trajectory tracking*.

- Pengujian ini dilakukan dengan penggunaan model kinematika (3.2) pada *trajectory* lingkaran dan persegi.
 - Hasil yang akan diukur yaitu kecepatan linier, kecepatan angular dan *error tracking*.
 - Perbandingan nilai *settling time error tracking* dari kontroler hasil perancangan dengan kontroler dari kajian pustaka baik pada *smooth trajectory* maupun *nonsmooth trajectory*.
3. Pengujian sistem kontrol dengan satu pengukuran untuk *trajectory tracking*
- Pengujian ini dilakukan dengan penggunaan model dinamika pada *trajectory* berupa garis lurus maupun persegi.
 - Hasil yang akan diukur yaitu *error tracking*.
 - Kriteria pengujian yaitu untuk *error tracking* pada garis lurus *settling time* konvergensi *error tracking* dapat tercapai dalam $t < 20s$ sedangkan pada *trajectory* yang lain *settling time* konvergensi *error tracking* dapat tercapai dalam $t < 30s$
4. Pengujian sistem kontrol dengan dua pengukuran untuk *trajectory tracking* tanpa gangguan
- Hasil yang akan diukur yaitu *error tracking*.
 - Kriteria pengujian pada *trajectory* segi delapan yaitu *settling time* konvergensi *error* dapat tercapai dalam $t < 30s$.
 - Kriteria pengujian pada *trajectory* persegi yaitu *settling time* konvergensi *error* dapat tercapai dalam $t < 30s$.
5. Pengujian sistem dengan dua pengukuran untuk *trajectory tracking* dengan gangguan eksternal berupa besi
- Pengujian dilakukan dengan memberikan gangguan pada salah satu *magnetic spot*. Jarak gangguan dari titik maksimum MFD akan diteliti sehingga *trajectory tracking* tetap memenuhi kriteria yang ditentukan.
 - Kriteria pengujian pada *trajectory* segi delapan yaitu *settling time* konvergensi *error* dapat tercapai dalam $t < 30s$.
 - Kriteria pengujian pada *trajectory* persegi yaitu *settling time* konvergensi *error* dapat tercapai dalam $t < 30s$.

Halaman ini sengaja dikosongkan

BAB 4

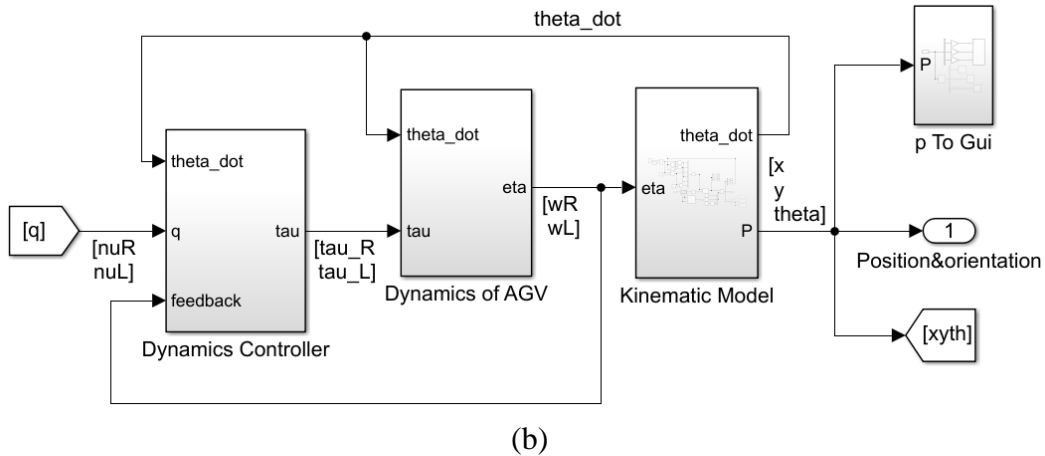
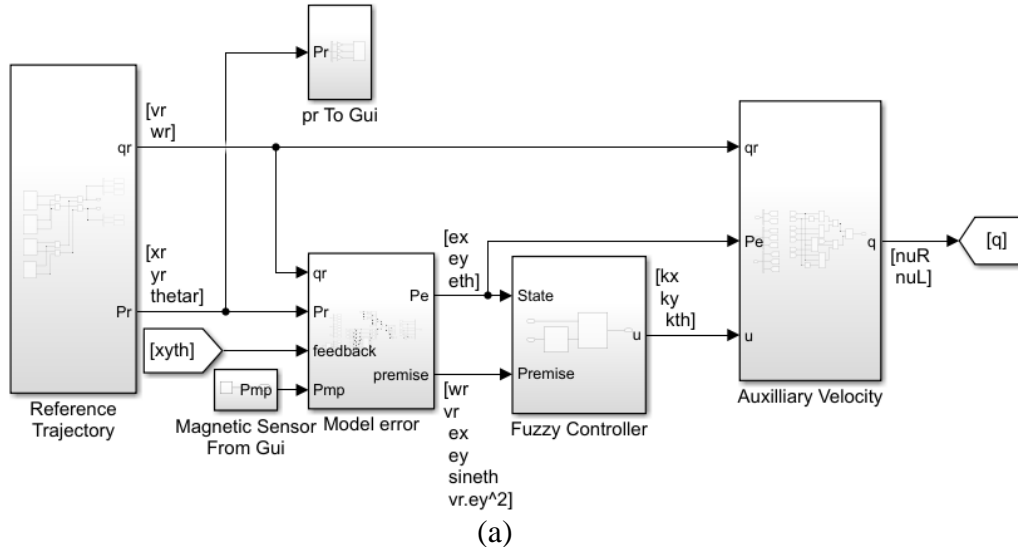
HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil simulasi dari perancangan sistem pada Bab 3 dimana simulasi dilakukan dengan menggunakan Simulink dan Matlab GUI baik untuk *plant* berupa kinematika maupun dinamika AGV kemudian dilakukan analisis data pengujian untuk mengetahui apakah hasil yang didapatkan sesuai dengan tujuan penelitian.

4.1 Model Simulink dari Kontroler AGV

Berdasarkan perancangan sistem kontrol pada Bab 3, penerapan pada SIMULINK telah dilakukan dengan menggunakan model kinematika (3.2) dan dinamika (3.15) untuk menguji hasil perancangan pada masing-masing *trajectory* referensi seperti yang ditunjukkan pada Gambar 4.1. Fuzzy Mamdani dibangun menggunakan blok fungsi, kontroler fuzzy T-S disusun berdasarkan desain PDC menggunakan blok fuzzy dan perhitungan *gain* dari LMI melalui M-File, dan kontroler dinamika menggunakan blok fungsi. Untuk mendukung pengujian secara *realtime*, animasi menggunakan MATLAB–Guide telah dilakukan dengan mengirim data dari SIMULINK ke GUI. Kemudian untuk data pengukuran digunakan blok model kinematika

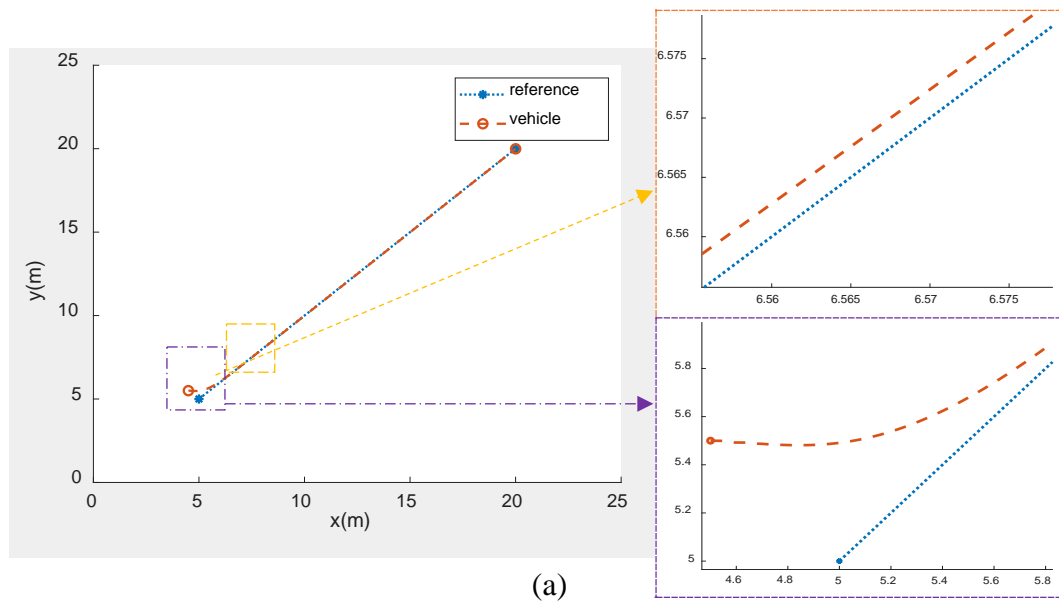
Pada Gambar 4.1 terlihat untuk perhitungan *input auxiliary velocity* pada blok *auxiliary velocity* menghasilkan *output* berupa kecepatan linier dan angular sedangkan *input* untuk kontroler dinamika adalah kecepatan angular roda kiri dan roda kanan sehingga dilakukan konversi berdasarkan (3.1). Untuk blok DOB terdapat pada model *error* yang digunakan pada pengujian menggunakan dinamika AGV dengan kontroler kinematika dan dinamika. *Nonlinear dynamics feedback* dan *forward gain* terdapat pada kontroler dinamika dimana *feedback* yang diberikan berupa kecepatan angular roda kanan dan roda kiri.



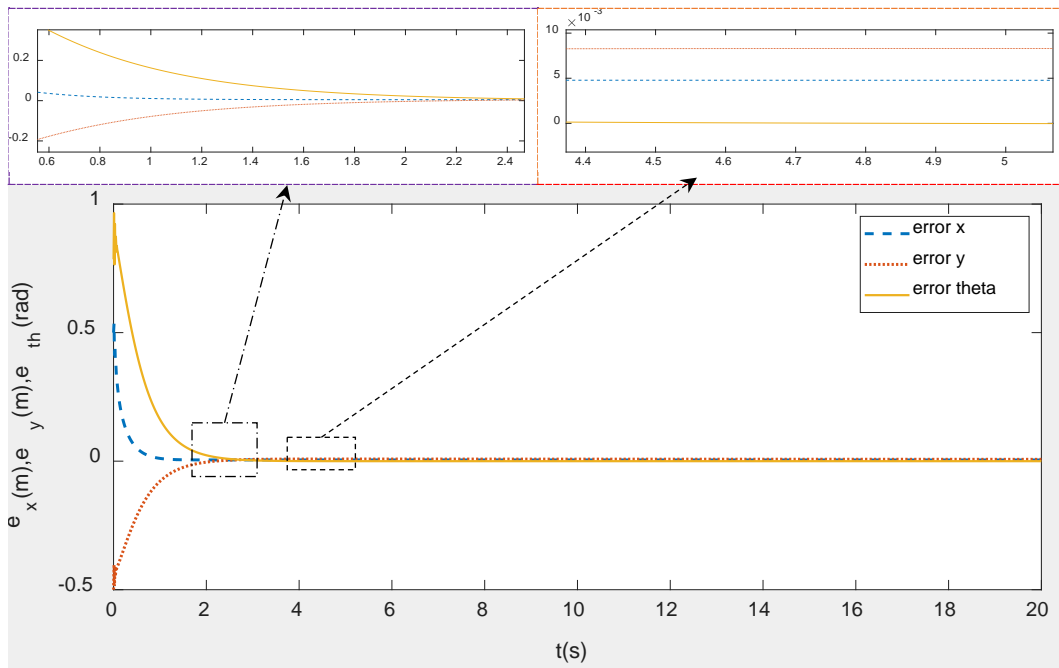
Gambar 4.1 Model Simulink dari Kontroler AGV (a) Kontroler Kinematika (b) Kontroler Dinamika

4.2 Pengujian Kontroler pada Model Kinematika

Pada pengujian ini digunakan *plant* berupa model kinematika AGV (3.2) dengan satu pengukuran. Pengujian pertama dilakukan pada 3 *trajectory* referensi yaitu garis lurus, segi delapan, dan persegi. Kontroler yang digunakan adalah kontroler fuzzy T-S pada sub-bab 3.3.2. Pengujian kedua dilakukan untuk membandingkan respon dari penggunaan kontroler fuzzy T-S yang dirancang pada tesis ini dengan kontroler dari metode sebelumnya yaitu kajian pustaka [11] dalam *trajectory tracking* garis lurus dan persegi. Pengujian ketiga dilakukan untuk menguji pengaruh kecepatan terhadap *trajectory tracking* pada lingkaran dan persegi.



(a)

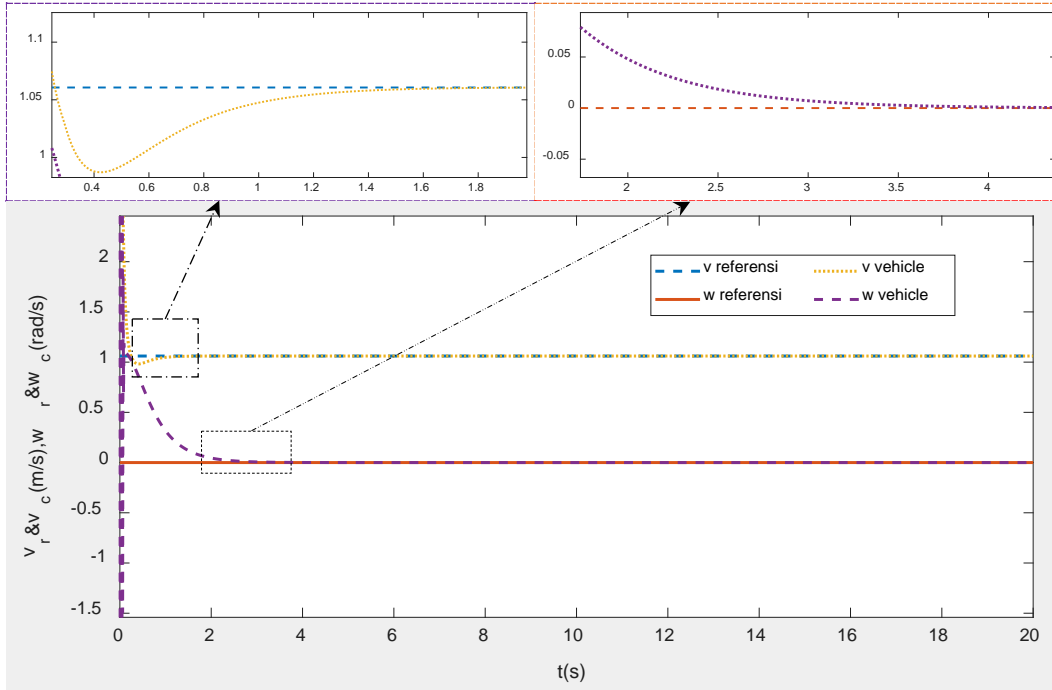


(b)

Gambar 4.2 (a) *Trajectory Tracking* Garis Lurus menggunakan Model Kinematika
(b) *Error Tracking*

4.2.1 Pengujian Kontroler Fuzzy T-S pada Model Kinematika

Kontroler yang digunakan pada pengujian ini sesuai dengan perancangan pada sub-bab 3.3.2 dengan model kinematika (3.2) sebagai *plant*. Pengujian dilakukan pada 3 *trajectory* referensi yang dirancang tanpa perubahan sudut, dengan perubahan sudut kecil, dan dengan perubahan sudut yang besar.



Gambar 4.3 Respon Kecepatan Linier dan Angular pada Garis Lurus

Hasil *tracking* pada *trajectory* garis lurus dengan inisial *error* (0.5, -0.5, 0.7854) dan kecepatan linier referensi (v_r) sebesar 1 m/s ditunjukkan pada Gambar 4.2. Pada Gambar 4.2b ditunjukkan tercapainya *steady state* pada semua *error* dengan hasil respon yang ditunjukkan pada Tabel 4.1.

Pada pengujian ini digunakan LMI pada (2.43), (3.43), (3.44), (3.49), dan (3.50).

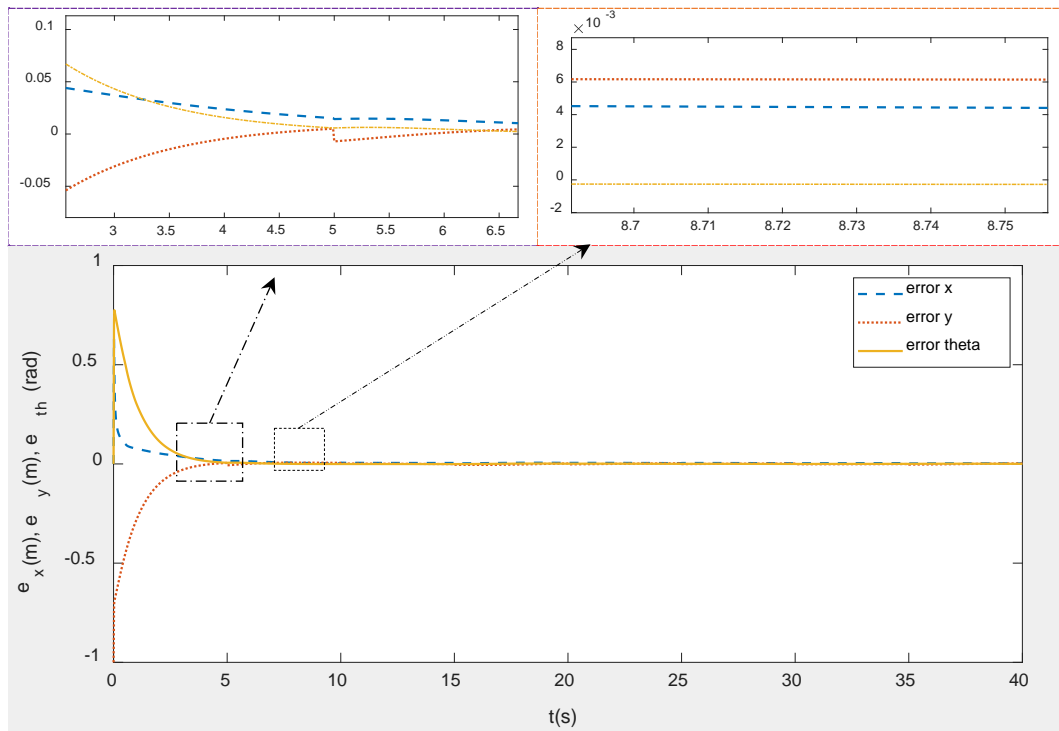
Matriks stabilitas P yang didapatkan dari LMI adalah sebagai berikut:

$$P_{ks} = \begin{bmatrix} 1.1496 & -0.1727 & -0.1005 \\ -0.1727 & 63.0547 & 30.1846 \\ -0.1005 & 30.1846 & 15.7346 \end{bmatrix}, eig(P_{ks}) = \begin{bmatrix} 1.0402 \\ 1.1511 \\ 77.74 \end{bmatrix}$$

Dengan demikian didapatkan matriks P positif definit yang memenuhi perancangan pada sub-bab 3.3.2. Program MATLAB-Script untuk perhitungan LMI dapat dilihat pada Lampiran C.5. Kemudian dari analisis *steady state error* matriks parameter Q (3.62) didapatkan sebagai berikut:

$$Q_{ks} = \begin{bmatrix} 11.6539 & -0.1321 & -0.1404 \\ -0.1321 & 15.0762 & -2.4120 \\ -0.1404 & -2.4120 & 11.5525 \end{bmatrix}$$

Untuk memastikan bahwa matriks parameter Q memenuhi (3.63) digunakan MATLAB-Script pada Lampiran C.6 dengan menggunakan kombinasi nilai $\bar{\epsilon}$ baik dengan nilai positif maupun negatif.

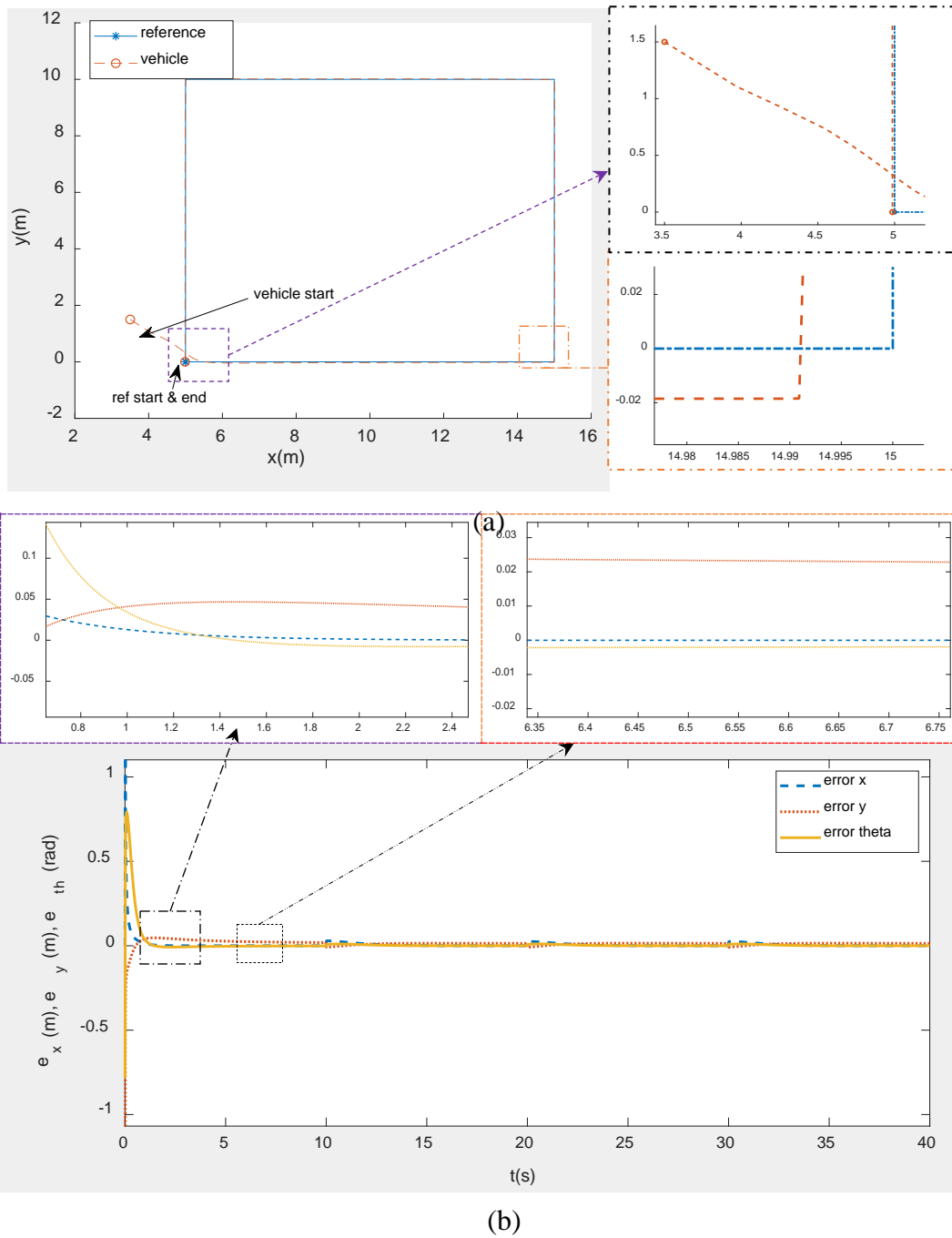


Gambar 4.4 *Error Tracking* pada *Trajectory* Segi Delapan

Dari pengujian pada garis lurus didapatkan fungsi Lyapunov yang memenuhi (3.63) yaitu $V(t) > 0$ dan $\dot{V}(t) < 0$ pada semua t . Nilai *decay rate* (3.49) yang dihasilkan adalah 0.9. Walaupun hasil yang didapatkan cukup bagus, kecepatan linier dan angular masih mengalami osilasi pada awal pergerakan AGV.

Pengujian kedua dilakukan pada referensi segidelapan dengan masing-masing sudut sebesar 0.7854 radian, inisial *error* (0, −1, 0), dan kecepatan linier dan angular referensi sebesar 1–1.41 m/s dan 0–354 rad/sec. Hasil *tracking* dapat dilihat pada Lampiran A.2 dan Tabel 4.1.

Nilai *settling time* pada semua *error* tercapai dibawah 10 s dengan toleransi *error* sebesar ± 0.01 seperti yang ditunjukkan pada Gambar 4.4. Hasil respon untuk *rise time* dari *error lateral* lebih lambat dikarenakan terjadinya *overshoot* pada *error* orientasi pada awal pergerakan yang mengakibatkan AGV berbelok ke kiri atau menjauhi referensi. Nilai variabel *premise* yang digunakan juga berpengaruh terhadap keterlambatan *rise time* ini. Ketika dilakukan pengurangan nilai variabel *premise error lateral* yang digunakan, nilai *rise time* menjadi cepat namun terjadi *peak* yang besar sehingga *settling time* jauh lebih lambat.



Gambar 4.5 (a) *Trajectory Tracking* pada Persegi menggunakan Model Kinematika
(b) *Error Tracking*

Untuk hasil pengujian pada *trajectory* segidelapan ditunjukkan pada Tabel 4.1, nilai MSE yang paling kecil yaitu *error leading* namun *steady state* yang dihasilkan paling besar disebabkan adanya *overshoot* kecil pada setiap belokan walaupun nilainya masih dibawah batas toleransi *error* yang dirancang sedangkan untuk *error* orientasi nilai *steady state* yang dihasilkan cukup kecil.

Tabel 4.1 Hasil Pengujian Kontroler pada Model Kinematika

Trajectory	Error	Satu Pengukuran	
		Settling time	MSE
Garis Lurus	e_x	0.8701	1.6×10^{-3}
	e_y	2.0322	4.1×10^{-2}
	e_θ	2.0892	1.3×10^{-2}
Segi Delapan	e_x	4.9915	9.01×10^{-4}
	e_y	3.3876	7.2×10^{-3}
	e_θ	3.9917	8.4×10^{-3}
Persegi	e_x	3.43	4.6×10^{-3}
	e_y	4.2	1.82×10^{-3}
	e_θ	2.0844	1.8×10^{-3}

Matriks stabilitas P didapatkan adalah sebagai berikut:

$$P_{kh} = \begin{bmatrix} 1.7118 & 0.0141 & 0.0138 \\ 0.0141 & 4.9507 & 2.0075 \\ 0.0138 & 2.0075 & 2.5092 \end{bmatrix}, \text{eig}(P_{kh}) = \begin{bmatrix} 2.9537 \\ 3.7594 \\ 30.4343 \end{bmatrix}$$

Pengujian ketiga dilakukan pada *trajectory* referensi persegi dengan sudut sebesar 1.5708 radian. Inisial *error* yang diberikan yaitu (0,1,0). Hasil *trajectory tracking* dapat dilihat pada Lampiran A.3.

Hasil pengujian untuk nilai inisial *error* yang berbeda ditunjukkan pada Lampiran A.4. Pada pengujian menggunakan model kinematika ini variabel *premise* yang memberikan pengaruh besar terhadap *steady state error* adalah z_6 ($v_r e_y^2$) dimana perubahan nilai mempengaruhi besar *overshoot error* orientasi yang terjadi pada setiap belokan seperti yang ditunjukkan pada Gambar 4.5. Kecepatan respon *settling time* dipengaruhi oleh nilai *output constraint* (3.46) yang digunakan, hal ini ditunjukkan pada Gambar 4.5 dimana ketika nilai *constraint* semakin besar nilai *settling time* yang didapatkan semakin kecil hingga titik maksimum yaitu 100 (pada Gambar 4.5a nilai λ sebesar 10).

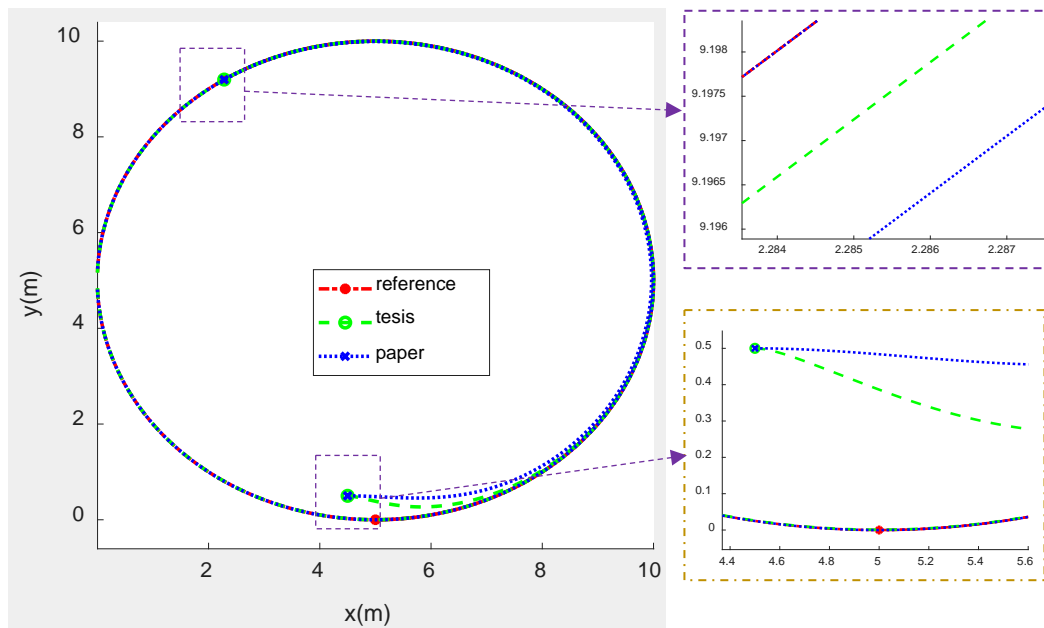
Tabel 4.2 Perbandingan *Settling Time Error* dari Hasil Perancangan dengan Kajian Pustaka [11] pada *Trajectory* Lingkaran

Pengujian	Hasil dari thesis			Hasil dari [11]		
	e_x	e_y	e_θ	e_x	e_y	e_θ
1	7.8788	4.4172	5.4836	27.0461	24.9055	13.0554
2	9.8323	4.5085	5.1092	17.3032	14.8133	8.4165
3	0.0717	2.2193	2.8392	10.5208	19.4924	10.0062
4	12.4768	8.9419	9.1896	5.2373	11.3467	12.8676
Rata-rata	7.5649	5.0217	5.6554	15.0268	17.6395	11.0864

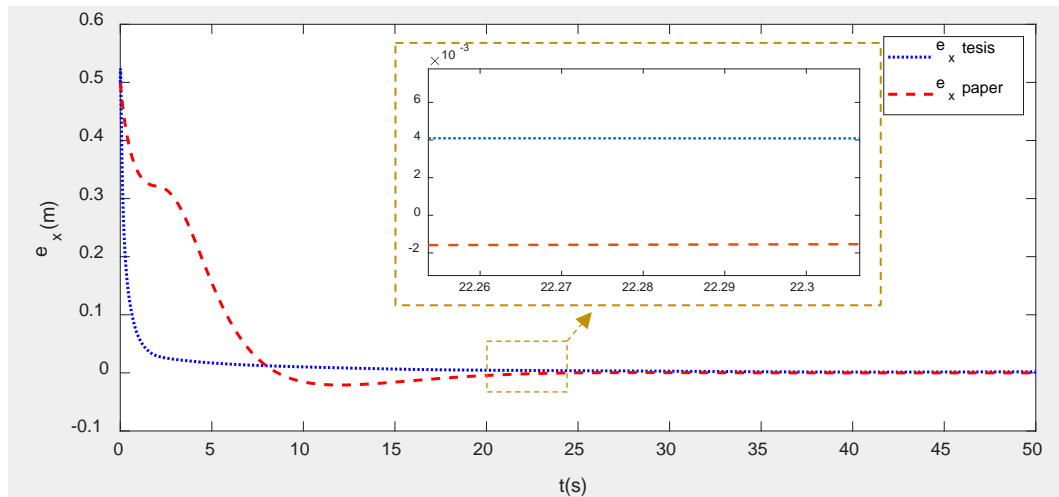
Berdasarkan pengujian yang telah dilakukan pada beberapa *trajectory* referensi dapat ditunjukkan bahwa kontroler fuzzy T-S yang dirancang dapat menghasilkan *steady state error* dan *settling time* sesuai dengan rencana pengujian seperti yang ditunjukkan pada Tabel 4.1. Kondisi kestabilan kontroler juga memenuhi syarat LMI pada (3.42), (3.43), (3.45), (3.48), dan (3.49) dengan didapatkannya matriks stabilitas P dan matriks parameter Q yang memenuhi (3.63) sehingga dapat disimpulkan (3.60) merupakan fungsi Lyapunov.

4.2.2 Pengujian Perbandingan *Trajectory Tracking* antara Hasil Perancangan dengan Metode dari Kajian Pustaka

Hasil *tracking smooth trajectory* dari [11] cukup bagus namun belum dilakukan pengujian pada *nonsmooth trajectory*. Untuk itu akan dilakukan pengujian menggunakan kontroler yang diajukan pada [11] untuk dibandingkan dengan kontroler hasil rancangan. *Trajectory* referensi yang digunakan yaitu *smooth trajectory* berupa lingkaran dan *nonsmooth trajectory* berupa persegi. Pada pengujian ini digunakan model kinematika (3.2) dan MATLAB-Script dapat dilihat pada Lampiran C.7. Untuk fuzzy T-S model yang digunakan dari tesis adalah (3.8) sedangkan dari metode sebelumnya [11] adalah (2.19). Hasil *error tracking* dari pengujian kontroler fuzzy T-S hasil rancangan dan metode sebelumnya untuk *smooth trajectory* berupa lingkaran ditunjukkan pada Gambar 4.6.

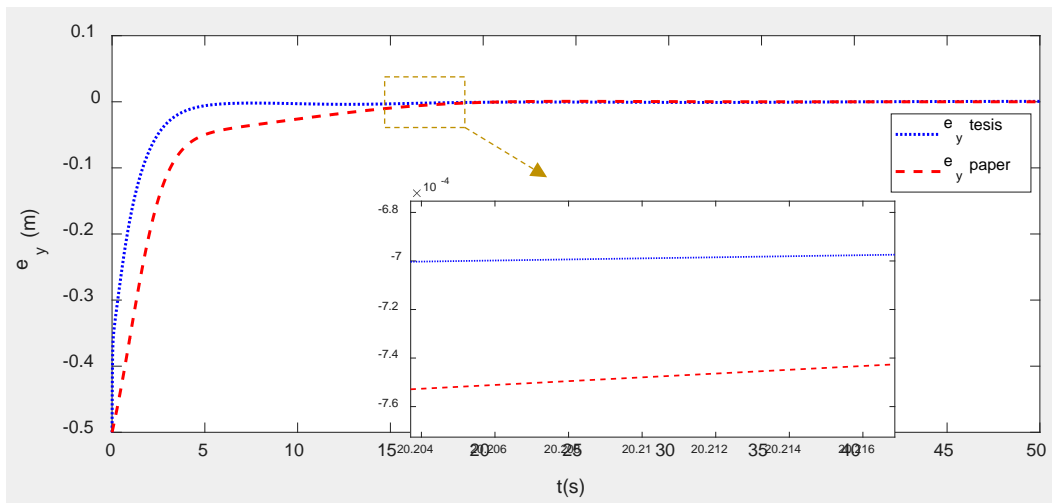


Gambar 4.6 Perbandingan *Tracking Error* pada *trajectory* Lingkaran

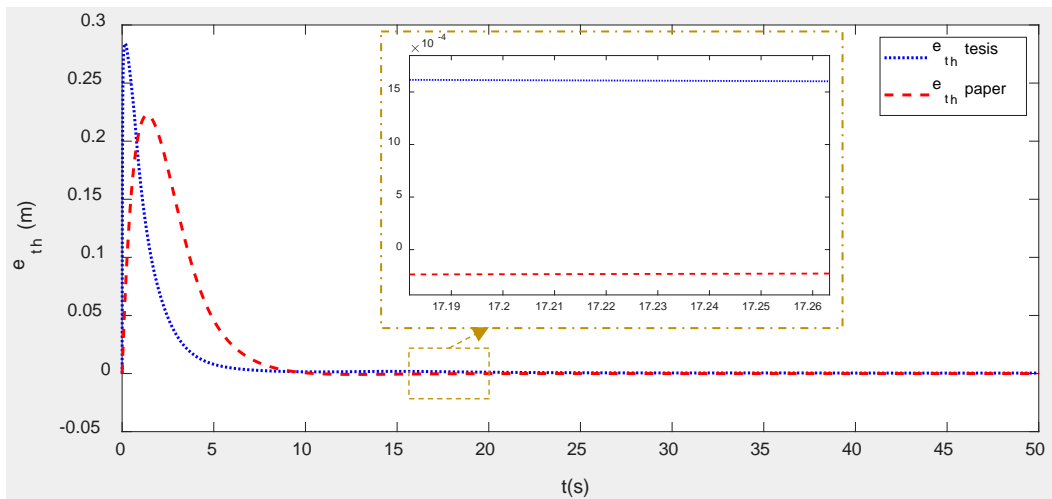


Gambar 4.7 Perbandingan Hasil *Tracking Error Leading* pada Lingkaran

Pada pengujian pertama dilakukan dengan v_r dan ω_r sebesar 1 m/s dan 0.2 rad/s dengan inisial *error* (0.5, -0.5, 0). Dari pengujian hasil rancangan kontroler didapatkan nilai *settling time* yang cukup kecil dan *steady state* tercapai untuk semua *error* (e_x , e_y , dan e_θ). Perbandingan *error leading* antara penggunaan kontroler dari [11] dan hasil rancangan tesis ditunjukkan pada Gambar 4.7 dimana *settling time* yang dihasilkan dari kajian lebih lambat yaitu 17 s akibat adanya *undershoot* sebesar 200%. Nilai *rise time* yang cukup lama yaitu 6.62 s juga mempengaruhi lambatnya *settling time* ini.

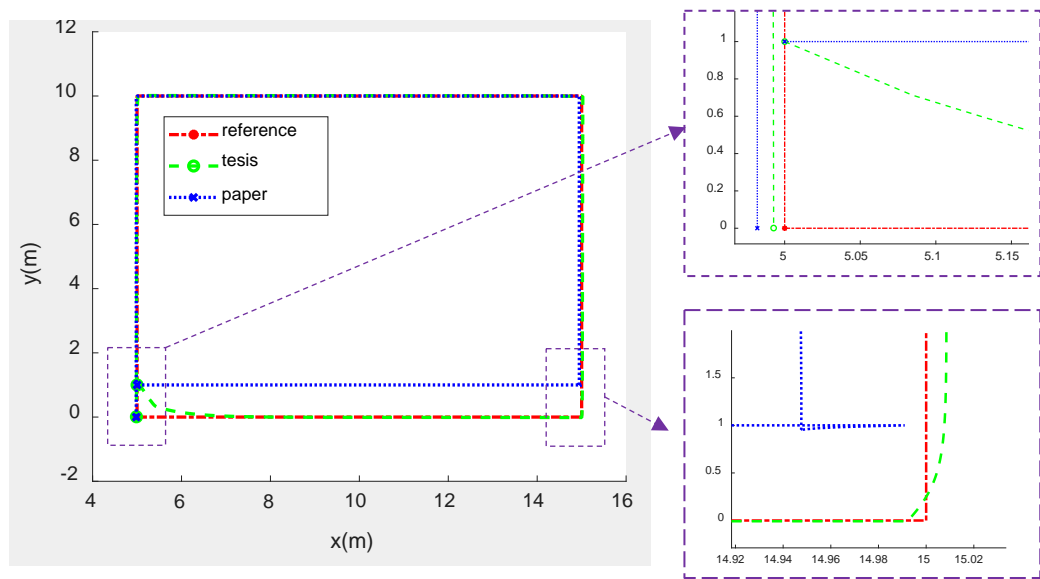


Gambar 4.8 Perbandingan Hasil *Tracking Error Lateral* pada Lingkaran

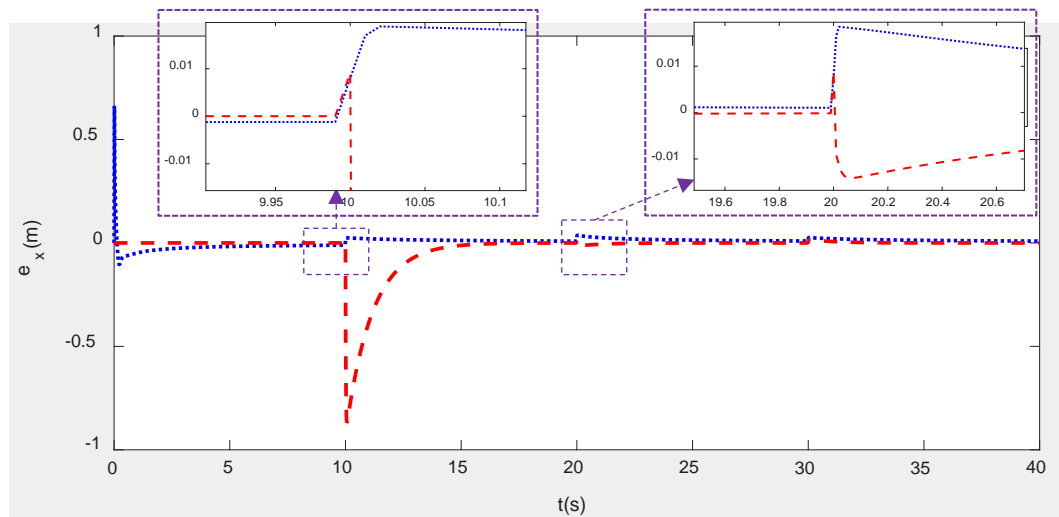


Gambar 4.9 Perbandingan Hasil *Tracking Error Orientasi* pada Lingkaran

Kelebihan dari kontroler [11] terletak pada nilai *steady error* yang dihasilkan pada *error leading* didapatkan sebesar 2×10^{-4} m sedangkan dari hasil rancangan sebesar 4×10^{-3} m. Hasil respon *error lateral* ditunjukkan pada Gambar 4.7 dimana *settling time* yang didapatkan dari kontroler hasil perancangan lebih baik begitu juga dengan hasil *settling time error* orientasi. Untuk hasil dari beberapa pengujian dengan variabel *premise* yang berbeda ditunjukkan pada Tabel 4.2 dan Lampiran A.4. Pada Tabel 4.2 ditunjukkan walaupun didapatkan *steady state error* nilai *settling time* dari hasil perancangan kontroler lebih baik dengan selisih yang cukup besar dibandingkan menggunakan kontroler dari [11].



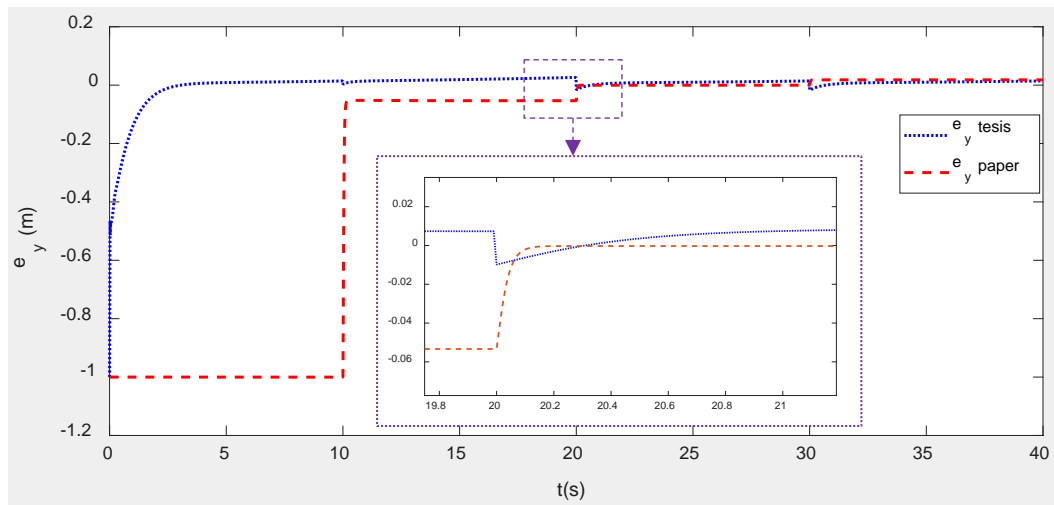
Gambar 4.10 Hasil *Tracking* pada Persegi [11]



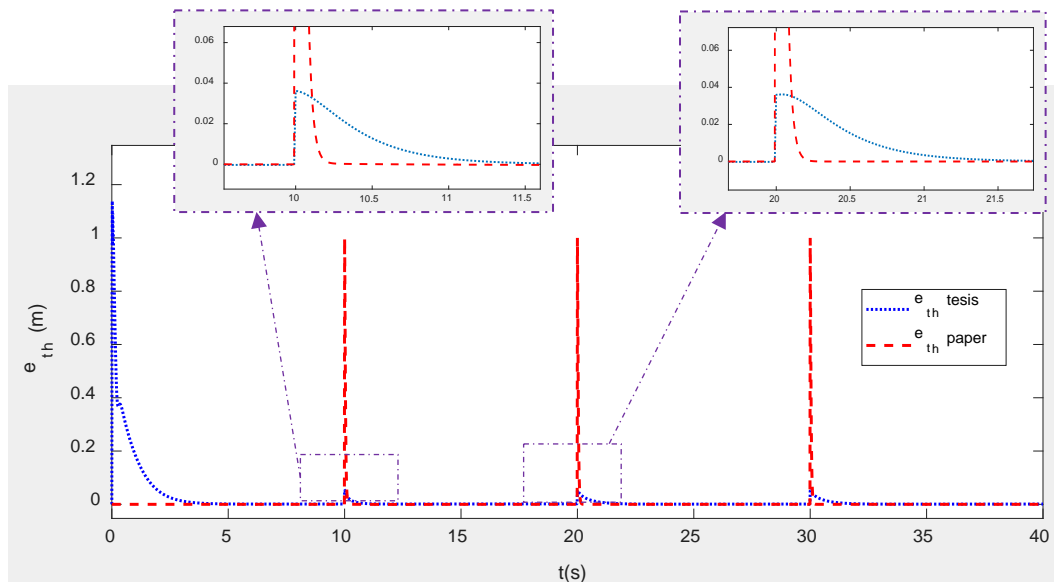
Gambar 4.11 Hasil Perbandingan *Tracking Error In Leading* pada Persegi

Hasil dari pengujian dapat dilihat pada Lampiran A.6. Hasil terbaik pengujian dengan perancangan kontroler pada tesis adalah pengujian 3 namun kecepatan awal q (v dan ω) yang dibutuhkan AGV cukup besar.

Pengujian selanjutnya dilakukan pada *nonsmooth trajectory* yaitu persegi. Dengan inisial *error* dan referensi kecepatan yang sama seperti pada pengujian 4.2.1 untuk *trajectory* persegi. Untuk hasil dari pengujian ditunjukkan pada Gambar 4.10 dengan menggunakan inisial *error* yang sama terlihat *settling time error tracking* cukup lama meskipun didapatkan matriks stabilitas P .



Gambar 4.12 Hasil Perbandingan *Tracking Error Lateral* pada Persegi



Gambar 4.13 Hasil Perbandingan *Tracking Error* Orientasi pada Persegi

Nilai *settling time error leading* dari hasil rancangan kontroler tesis lebih baik dibandingkan dengan hasil dari kontroler [11] karena pada belokan pertama terjadi *undershoot* sebesar 150%. Hal ini juga terjadi pada *error lateral* maupun *error orientasi* dimana pada setiap belokan terjadi keterlambatan respon dari kontroler akibat adanya perubahan yang besar dari sudut *trajectory* referensi sehingga dengan kecepatan linier yang konstan dibutuhkan kecepatan angular yang besar dari AGV dalam melakukan belokan. Dengan demikian dapat disimpulkan untuk kontroler hasil rancangan menghasilkan performa yang lebih baik dibandingkan metode sebelumnya terutama untuk nilai *settling time* yang dihasilkan.

Tabel 4.3 Pengujian Pengaruh Kecepatan terhadap *Trajectory Tracking* pada Lingkaran

Kecepatan (m/s)	<i>Settling Time</i> (s)		
	e_x	e_y	e_θ
1.5	6.5437	4.5204	5.3436
3	4.9746	2.1349	2.4844
4	3.7017	3.0585	3.0602

Tabel 4.4 Pengujian Pengaruh Kecepatan terhadap *Trajectory Tracking* pada Persegi

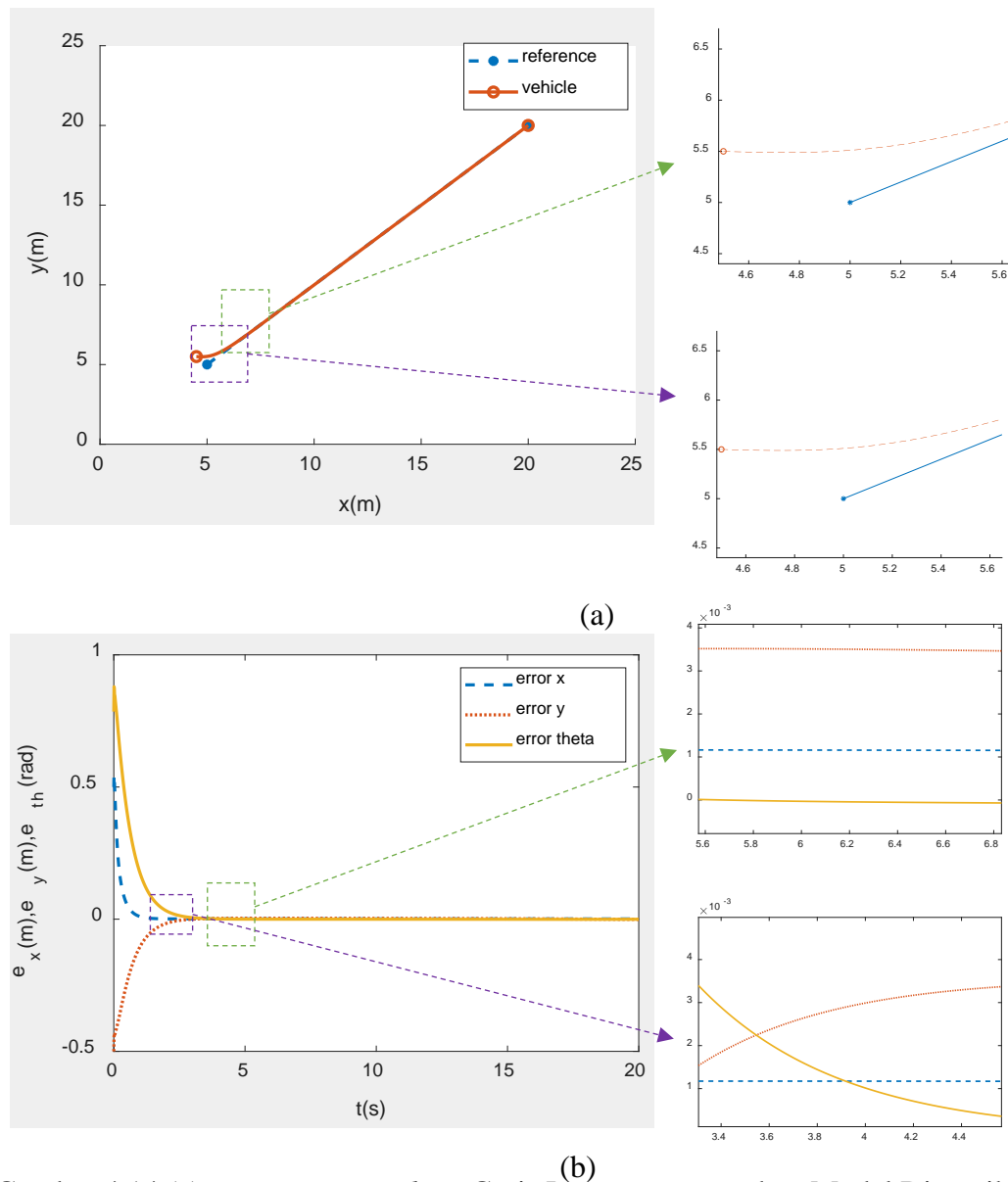
Kecepatan (m/s)	<i>Settling Time</i> (s)		
	e_x	e_y	e_θ
1.42	9.9917	7.9972	7.8703
2	15.0945	15.4990	15.5998
4	21.8584	1.2666	21.5239

4.3 Pengujian Pengaruh Kecepatan Terhadap Respon Tracking

Pada pengujian ini diberikan nilai referensi kecepatan yang berbeda pada *trajectory* lingkaran dan persegi seperti yang ditunjukkan pada Tabel 4.3. Inisial *error* diatur sama untuk setiap pengujian untuk lingkaran (0.5,-0.5,0) dan pada persegi (0,1,0). Berdasarkan Tabel 4.3 dan 4.4 dapat diketahui untuk pemberian kecepatan yang berbeda berpengaruh terhadap respon *settling time* yaitu semakin rendah kecepatan semakin lama *settling time* yang dihasilkan.

Dari setiap pengujian didapatkan matriks stabilitas P dan matrik parameter Q yang ditunjukkan oleh Lampiran A.8 walaupun terdapat *overshoot* pada setiap belokan pada *trajectory* persegi namun nilainya tidak terlalu besar yaitu 0.01 m sehingga masih dalam batas yang direncanakan.

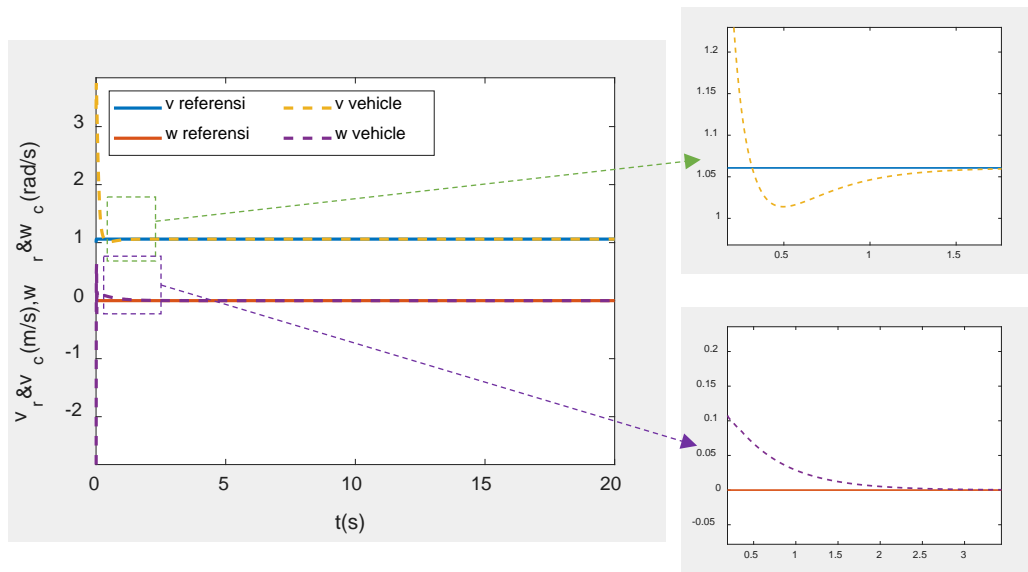
Respon yang dihasilkan pada pengujian *trajectory* lingkaran yaitu ketika pemberian kecepatan bertambah, nilai *settling time* pada *error tracking* semakin baik seperti yang ditunjukkan pada Tabel 4.3. Hal ini dikarenakan pada *smooth trajectory* terdapat perubahan sudut yang besar sehingga AGV dapat melakukan *tracking* dengan baik berbeda dengan *trajectory* persegi dimana dengan bertambahnya kecepatan, *settling time* yang didapatkan justru semakin lambat.



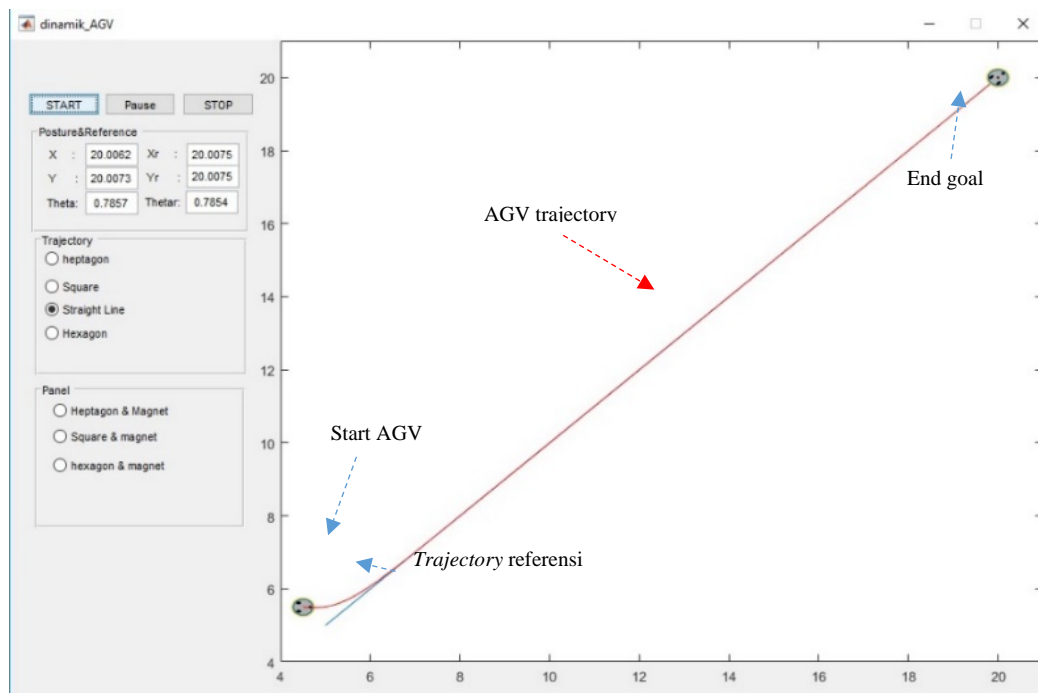
Gambar 4.14 (a) *Trajectory Tracking* Garis Lurus menggunakan Model Dinamika, (b) *Error Tracking*

4.4 Pengujian Kinematika dan Dinamika Kontroler pada Model Dinamika dengan Satu Pengukuran

Pada pengujian ini digunakan *plant* berupa dinamika AGV (3.15) dengan satu pengukuran. Pengujian dilakukan pada 4 *trajectory* referensi yaitu garis lurus, segidelapan, persegi, dan segienam. Kontroler yang digunakan adalah kontroler fuzzy T-S dan *nonlinear dynamics feedback*.



Gambar 4.15 Kecepatan Linier dan Angular *Trajectory Tracking* Garis Lurus



Gambar 4.16 Tampilan dari Matlab GUI untuk *Trajectory* Garis Lurus

Pengujian pertama dilakukan pada *trajectory* referensi berupa garis lurus menggunakan *plant* berupa model dinamika (3.15). Model ini merepresentasikan dinamika AGV sebagai fungsi dari kecepatan angular roda kanan dan roda kiri ($\dot{\phi}_R, \dot{\phi}_L$), kecepatan angular AGV ($\dot{\theta}$), dan torsi motor penggerak (τ_R, τ_L). Dengan parameter AGV yaitu $r = 0.025$ m, $b = 0.1$ m, $d = 0.05$ m, $m_c = 3$ kg, $m_w = 0.01$ kg, $I_c = 1.5625$ kg.m², dan $I_m = 0.00025$ kg.

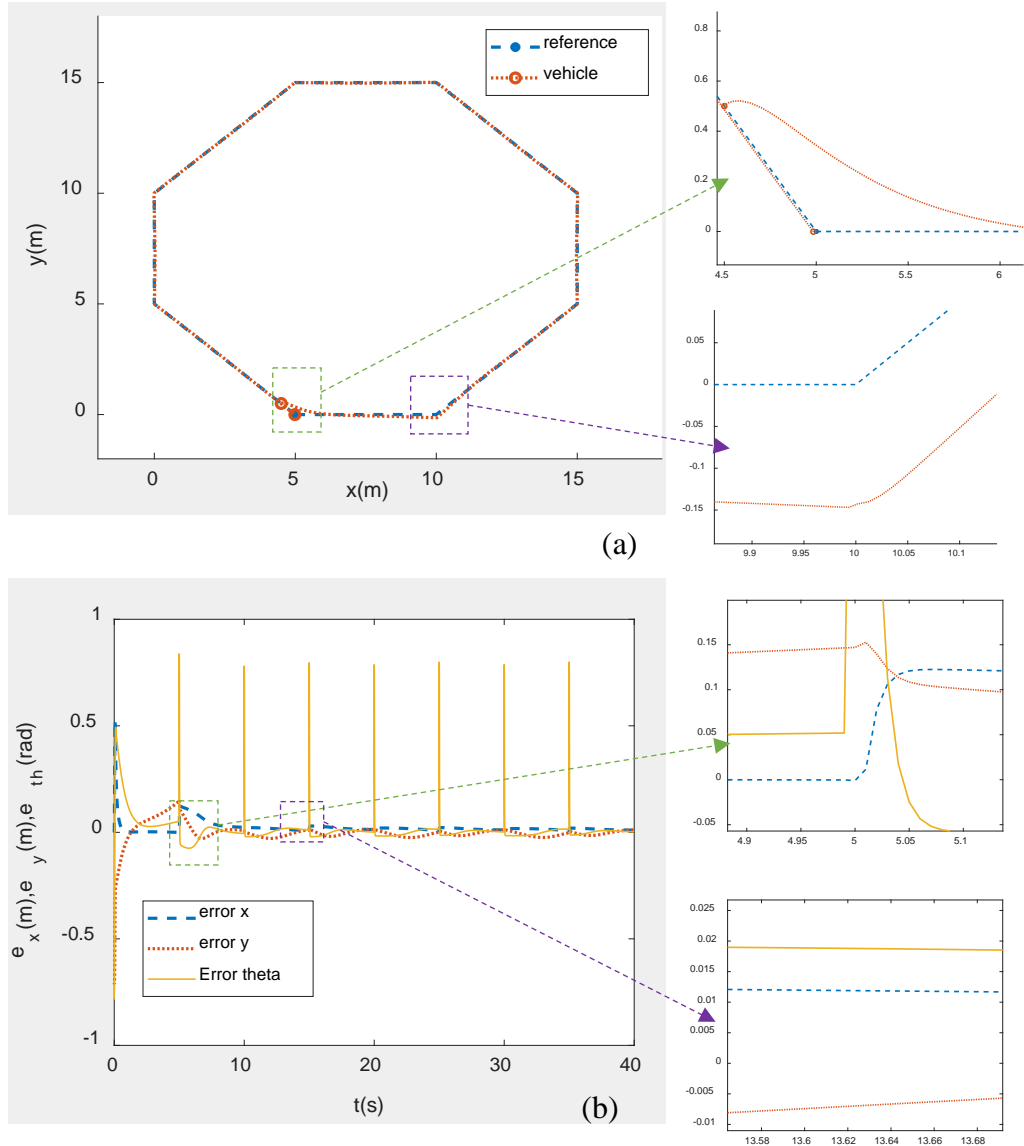
Pada Gambar 4.14 ditunjukkan hasil *tracking* pada *trajectory* garis lurus dengan *inisial error* (0.5,-0.5,0.7854) dan kecepatan linier referensi (v_r) sebesar 1 m/s. Hasil MSE dari masing-masing *error* yaitu *error leading*, *lateral*, dan orientasi yaitu 1.6×10^{-3} m, 4.1×10^{-2} , dan 1.3×10^{-2} . Walaupun hasil MSE tidak lebih baik dari *trajectory tracking* menggunakan model kinematika namun nilai *steady state* yang dihasilkan dari pengujian ini didapatkan hasil yang lebih baik. *Settling time* untuk *error leading* tercapai pada 0.9438 s, *error lateral* pada 2.18 s, dan *error orientasi* pada 2.35 s.

Kecepatan linier dan angular dari *input auxiliary velocity* mencapai kondisi *steady state* dengan nilai e_v sebesar 0 m/s dan e_ω sebesar 0 rad/sec seperti yang ditunjukkan pada Gambar 4.15. *Decay rate* yang dihasilkan sebesar 0.9 dengan matrik stabilitas P sebagai berikut:

$$P_{ds} = \begin{bmatrix} 3.7458 & 3.9356 & 1.3644 \\ 3.9356 & 65.5504 & 35.7439 \\ 1.3644 & 35.7439 & 22.5582 \end{bmatrix}, eig(P_{ds}) = \begin{bmatrix} 2.0845 \\ 4.8581 \\ 22.3885 \end{bmatrix}$$

Untuk pengujian kedua dilakukan pada *trajectory* segi delapan dengan sudut sebesar 0.7854 radian dan *inisial error* (0.5,-0.5,0.7854). Kecepatan linier referensi (v_r) yang digunakan sebesar 1 m/s dan kecepatan angular (ω_r) sebesar 0-350 rad/s. Hasil MSE dari masing-masing *error* yaitu *error leading*, *lateral*, dan orientasi didapatkan sebesar 4.3×10^{-3} m, 1.4×10^{-3} m, dan 2×10^{-3} radian.

Settling time untuk *error leading* tercapai pada 16 s, *error lateral* pada 15 s, dan *error orientasi* pada 35 s. Dari Gambar 4.17 terlihat bahwa *error orientasi* mengalami kenaikan pada setiap belokan dengan nilai 0.67 radian (38°), hal ini disebabkan oleh perubahan sudut dari *trajectory* terlalu besar. Pada detik ke 4.99, sudut orientasi referensi berubah dari 0 ke 0.7854 radian atau 78.54 rad/s untuk kecepatan angular referensi sedangkan AGV baru berbelok pada detik ke 5 sehingga terjadi *overshoot* yang besar. Namun pada detik ke 5.2, AGV sudah bisa melakukan *tracking* dengan baik dan hal ini terjadi disetiap belokan dengan selisih waktu yang sama.

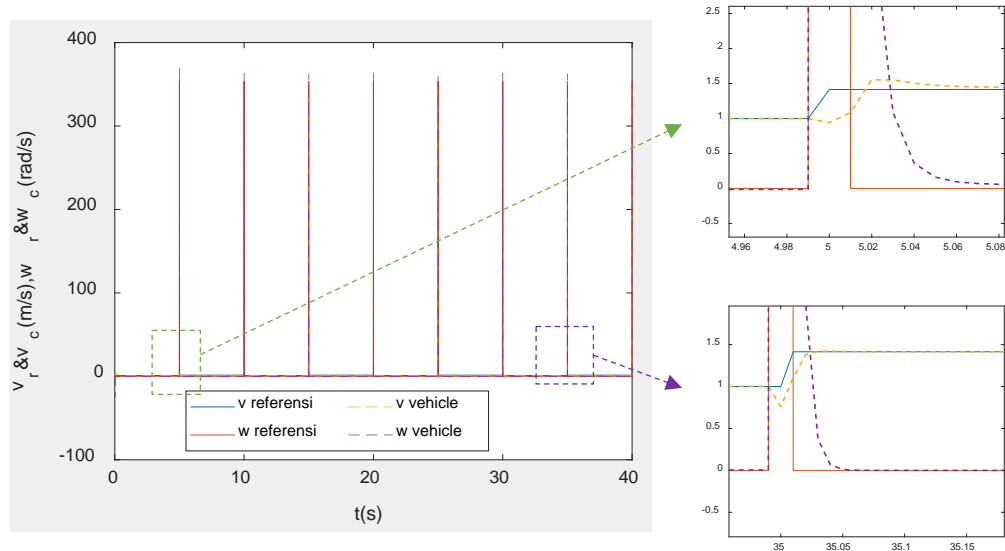


Gambar 4.17 *Tracking Trajectory* Segidelapan menggunakan Model Dinamika, (b) *Error Tracking*

Untuk MSE dari *error* kecepatan linier dan angular *input auxiliary velocity* yaitu sebesar 6.94×10^{-2} m/s dan 1.03 rad/sec. Untuk variabel *premise* dan parameter *gain* dinamika kontroler didapatkan sebagai berikut:

$$z = [1 \quad 1.4142 \quad 0.6 \quad 0.9 \quad 1 \quad 0.05]$$

$$\bar{k} = \begin{bmatrix} k_r \\ k_p \\ k_d \end{bmatrix} = \begin{bmatrix} 0.01 \\ 40 \\ 0.01 \end{bmatrix}$$

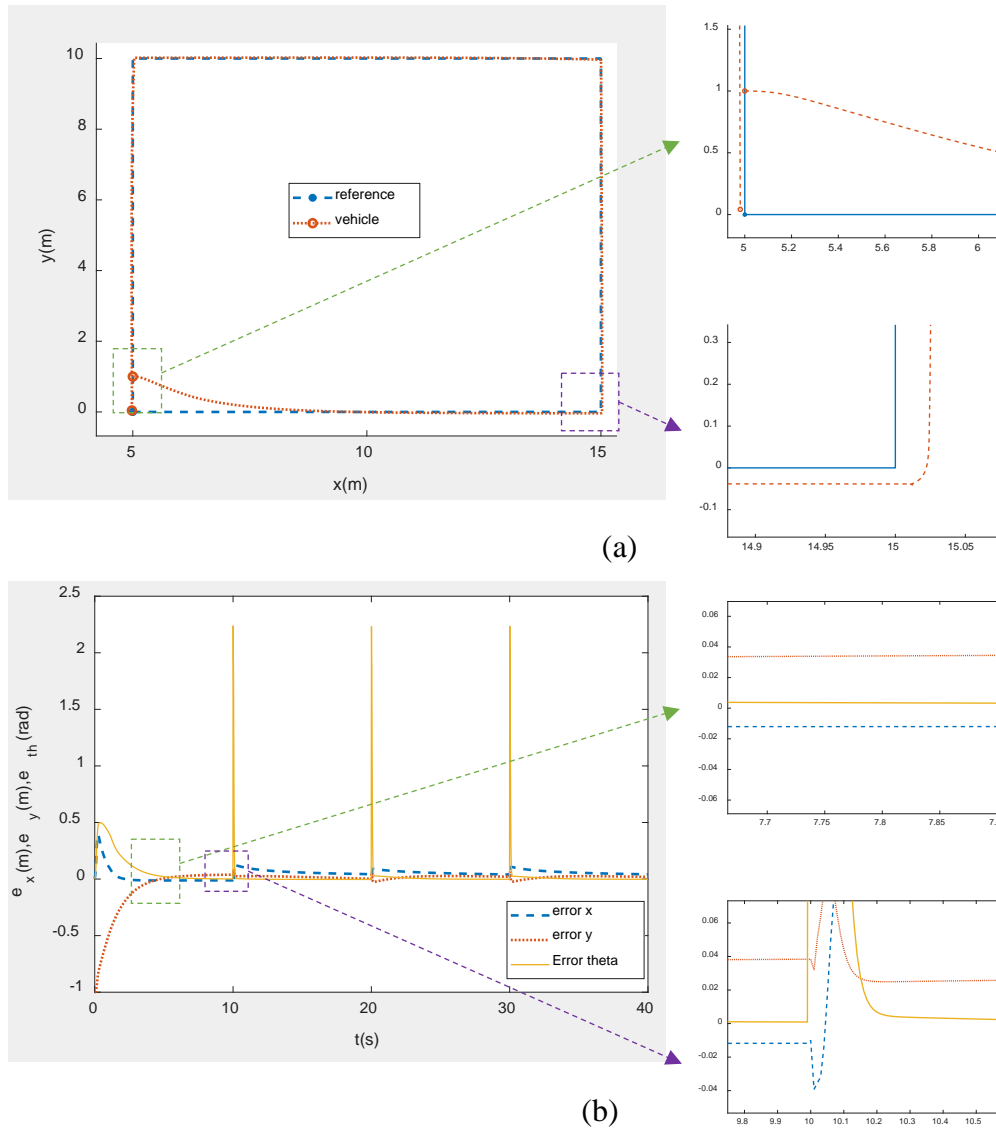


Gambar 4.18 Kecepatan Linier dan Angular *Trajectory Tracking Segidelapan*

Pengujian selanjutnya yaitu pada *trajectory* referensi persegi dengan sudut sebesar 1.5708 radian dan inisial *error* yang diberikan yaitu (0,1,0). Kecepatan linier dan kecepatan angular referensi sebesar 0.5–1 m/s dan 0–160 rad/sec. Hasil MSE *error tracking* pada pengujian ketiga adalah *error leading* sebesar 4.6×10^{-3} m, *error lateral* 1.82×10^{-2} m, *error orientasi* sebesar 1.8×10^{-3} radian. Nilai *settling time* pada *error leading* dan *lateral* dibawah 7 s dengan toleransi *error* sebesar ± 0.02 sedangkan untuk *error orientasi* 31s. Untuk MSE dari *error* kecepatan linier dan angular *input auxiliary velocity* yaitu sebesar 6.5×10^{-3} m/s dan 7.12×10^{-2} rad/sec.

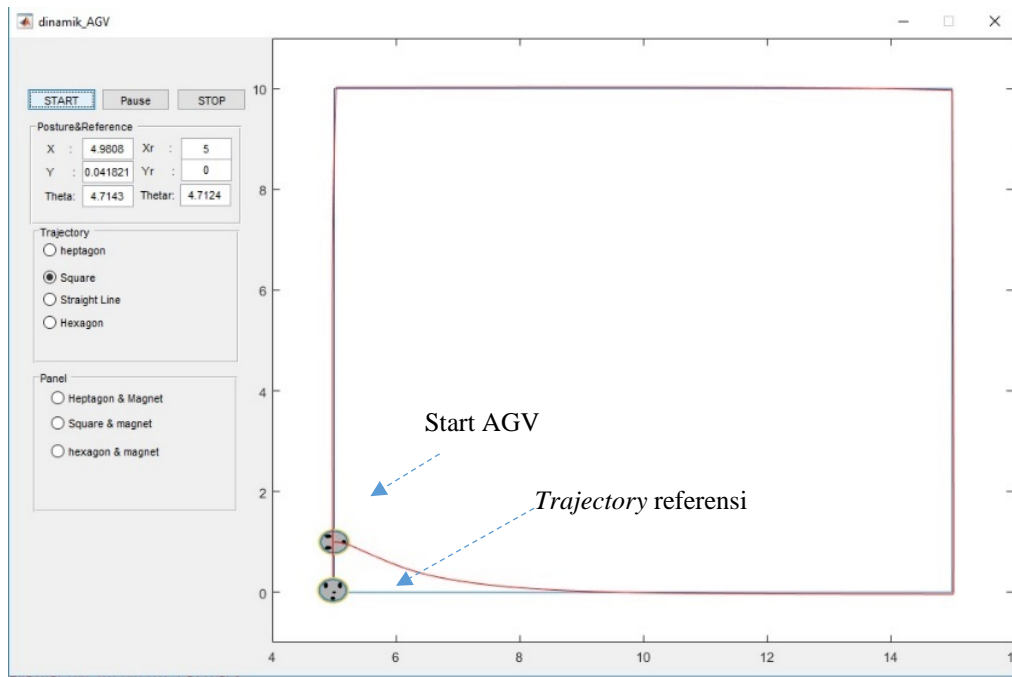
Pengujian keempat dilakukan pada *trajectory* segienam dengan variasi sudut antara 0.7854 hingga 1.5708 radian. Kecepatan linier dan angular referensi yaitu 0.58–1.42 m/s dan 0–300 rad/s. Nilai *inisial error* yang diberikan yaitu (0.5,−0.5,0.7854). *Settling time* untuk *error leading* tercapai pada 5 s, *error lateral* pada 10 s, dan *error orientasi* pada 36 s.

Hasil MSE dari masing-masing *error* yaitu *error leading*, *lateral*, dan orientasi didapatkan sebesar 2.8×10^{-3} m, 5.2×10^{-3} m, dan 8.4×10^{-3} radian. Sedangkan MSE dari *error* kecepatan linier dan angular *input auxiliary velocity* yaitu sebesar 5.52×10^{-2} m/s dan 0.288 rad/sec.



Gambar 4.19 (a) *Tracking Trajectory* Persegi menggunakan Model Dinamika, (b) *Error Tracking*

Berdasarkan pengujian yang telah dilakukan pada beberapa *trajectory* untuk *plant* berupa dinamika AGV dengan menggunakan kontroler kinematika dan dinamika, Untuk kinematika kontroler pada kontroler fuzzy T-S, kontroler yang dirancang telah memenuhi syarat stabilitas dimana kondisi kestabilan asimptotik dari kontroler ditunjukkan oleh terdapatnya matriks positif definit P yang sesuai dengan LMI (3.42), (3.43), (3.45), (3.48), dan (3.49). Kemudian untuk *input auxiliary velocity* didapatkan MSE yang kecil dengan toleransi 0.05 m/s untuk kecepatan linier dan 10 rad/s untuk kecepatan angular.



(b)

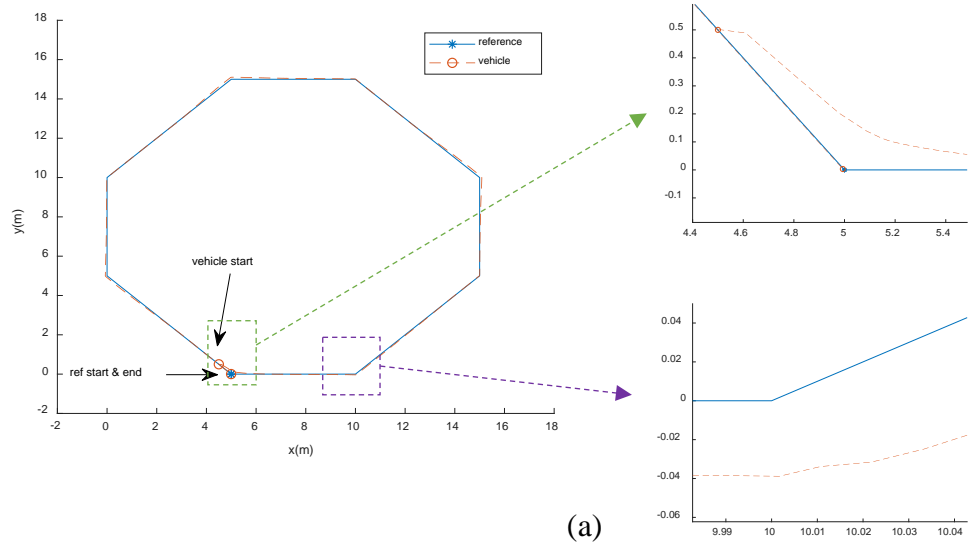
Gambar 4.20 Hasil *crop* dari Matlab GUI untuk *Trajectory* Persegi

Sedangkan pada kontroler dinamika, *forward gain* (k_r) dengan nilai yang kecil dapat mengurangi nilai *error* yang terjadi baik untuk kecepatan angular roda kanan maupun kiri yang terlihat dari konvergensi *error leading*, *lateral*, dan orientasi pada masing-masing pengujian. Namun untuk *settling time* pada *error* orientasi yang dihasilkan cukup lambat akibat perbedaan titik belok dari *trajectory* referensi dan AGV sehingga diperlukan sensor magnet untuk mengatasi hal ini.

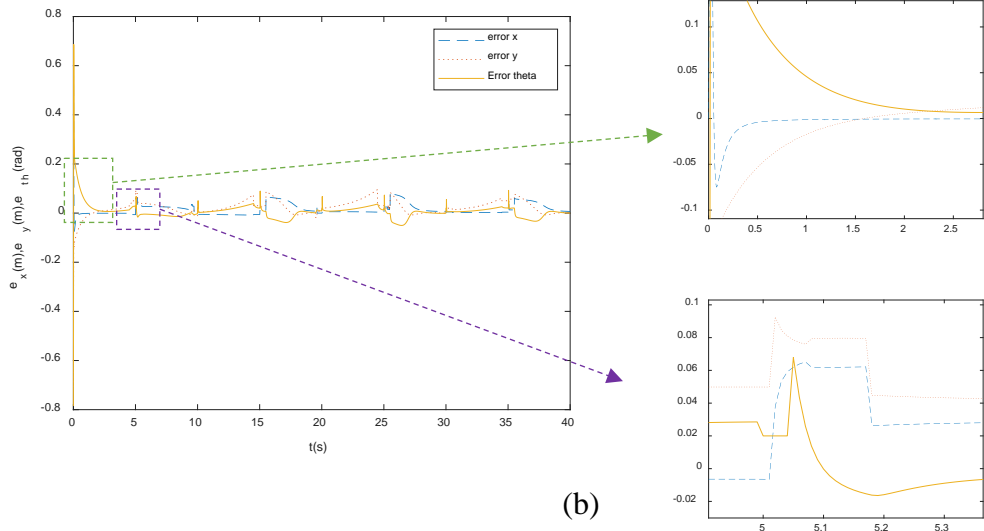
4.5 Pengujian *Tracking Trajectory* untuk Sistem dengan Dua Pengukuran

Pada pengujian ini digunakan *plant* berupa dinamika AGV dengan dua pengukuran yaitu sensor jarak untuk pengukuran kontinu dan sensor magnet untuk pengukuran MFD pada area magnet. Pengujian dilakukan pada 3 *trajectory* referensi yaitu segidelapan, persegi, dan segienam. Kontroler yang digunakan adalah kontroler fuzzy T-S dengan koreksi *error* menggunakan DOB dan *nonlinear dynamics feedback*.

Untuk pengujian kedua dilakukan pada *trajectory* segi enam dengan sudut sebesar 0.7854 radian dan *inisial error* (0.5, -0.5, 0.7854). *Settling time* untuk *error leading* tercapai pada 16 s, *error lateral* pada 15 s, dan *error orientasi* pada 35 s.



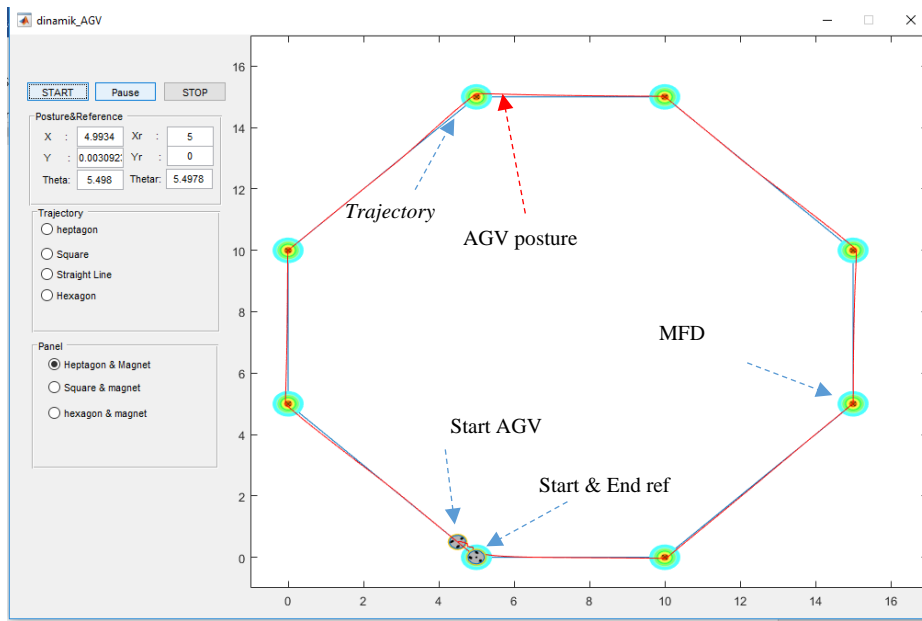
(a)



(b)

Gambar 4.21 (a) *Tracking* pada *Trajectory* Segidelapan menggunakan Model Dinamika, (b) *Error Tracking*

Kecepatan linier referensi (v_r) yang digunakan sebesar 1–1.5 m/s dan kecepatan angular (ω_r) sebesar 0–350 rad/s. Hasil MSE dari *error leading*, *lateral*, dan orientasi didapatkan sebesar 7.9×10^{-4} m, 1.5×10^{-3} m, dan 1.3×10^{-3} radian sedangkan MSE dari *error* kecepatan linier dan angular *input auxiliary velocity* yaitu sebesar 0.438 m/s dan 7.225 rad/sec.



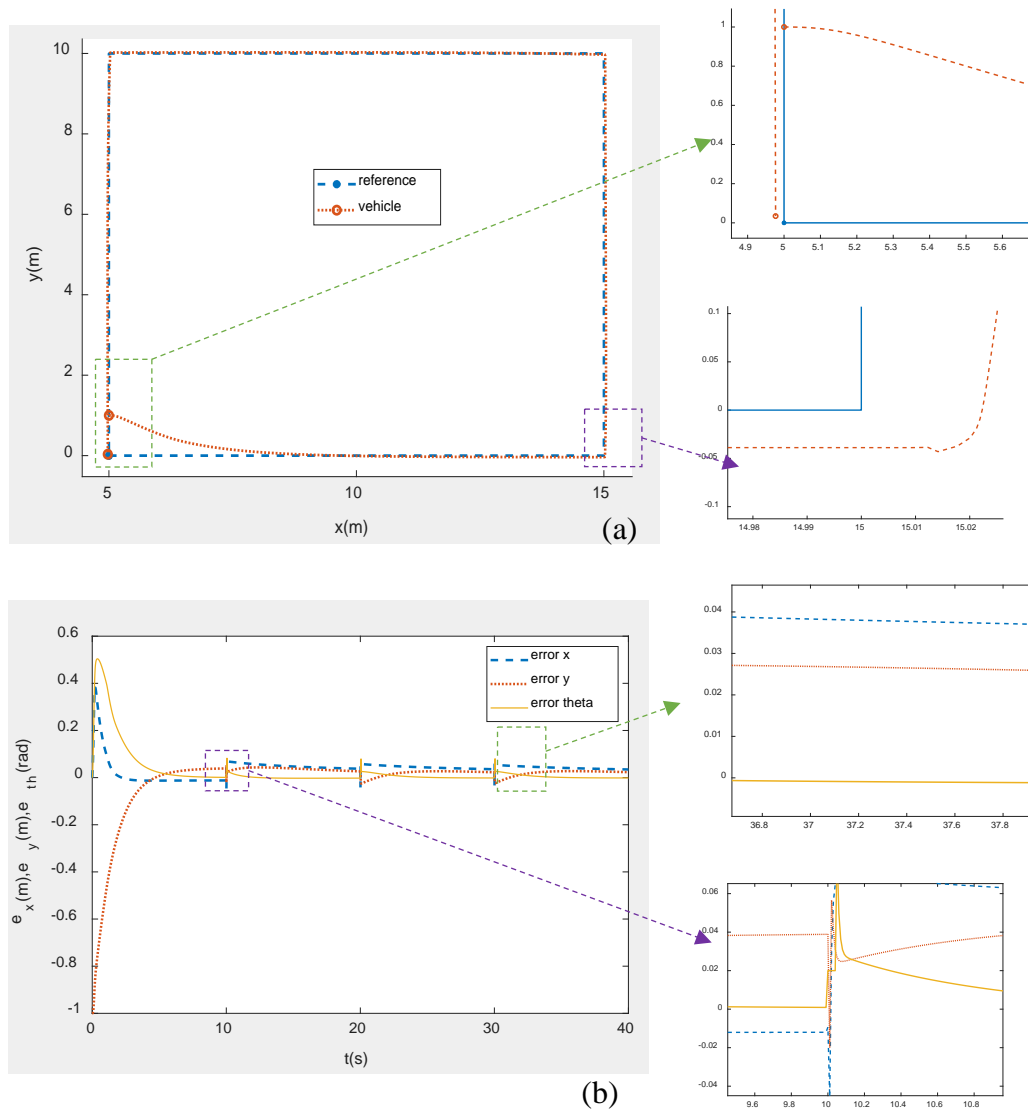
Gambar 4.22 Hasil *crop* Simulasi pada Matlab GUI

Untuk hasil dari MATLAB GUI ditunjukkan oleh Gambar 4.22 dengan plot MFD berdasarkan metode *numerical* FEA menggunakan software FEMM. Pada pengujian selanjutnya, kecepatan linier dan kecepatan angular referensi adalah 0.5–1 m/s dan 0–160 rad/sec. Dengan *inisial error* yang diberikan yaitu (0,1,0). Hasil MSE *error tracking* pada pengujian ketiga adalah *error leading* sebesar 3.2×10^{-3} m, *error lateral* sebesar 1.82×10^{-2} m, *error* orientasi sebesar 7.6×10^{-3} radian. Nilai *settling error* tercapai saat $t = 5$ s dengan toleransi *error* sebesar ± 0.1 . Nilai *decay rate* yang dihasilkan sebesar 0.05 dan parameter untuk kontroler kinematika dan dinamika adalah sebagai berikut:

$$P_{dsq} = \begin{bmatrix} 2.1568 & 0 & 0.0002 \\ 0 & 2.1568 & 0.0081 \\ 0.0002 & 0.0081 & 1.9695 \end{bmatrix}, eig(P_{dsq}) = \begin{bmatrix} 1.9691 \\ 2.1568 \\ 2.1571 \end{bmatrix}$$

$$z = [20 \quad 1 \quad 0.1 \quad 0.1 \quad 1 \quad 0.1]$$

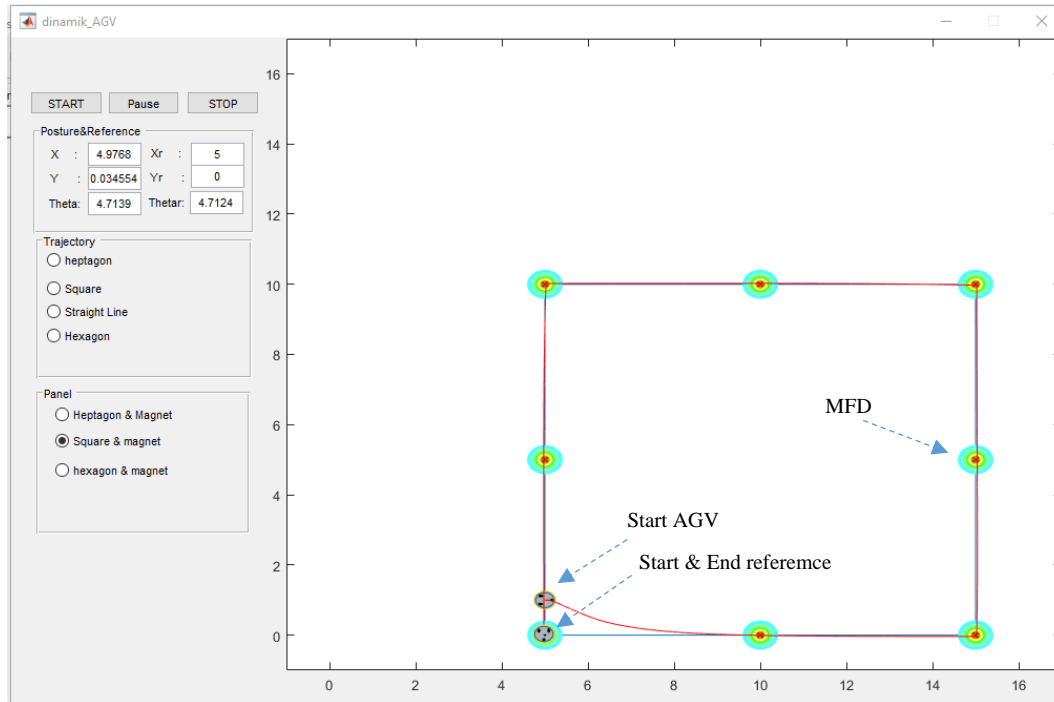
Pengujian selanjutnya dilakukan pada *trajectory* segi enam dengan sudut 0.7854–1.5708 radian. Kecepatan linier dan angular referensi yaitu 0.58–1.42 m/s dan 0–300 rad/s dan *inisial error* yang diberikan yaitu (0.5,−0.5,0.7854).



Gambar 4.23 (a) *Trajectory Tracking Persegi dengan Dua Pengukuran* (b) *Error Tracking*

Settling time untuk *error leading* tercapai pada 5 s, *error lateral* pada 10 s, dan *error orientasi* pada 36 s. *Decay rate* yang dihasilkan sebesar 1.6 dan parameter matriks P sebagai berikut:

$$P_{dhx} = \begin{bmatrix} 10.9828 & 4.1040 & 2.8281 \\ 4.1040 & 19.6094 & 3.9287 \\ 2.8281 & 3.9287 & 3.1529 \end{bmatrix}, eig(P_{dhx}) = \begin{bmatrix} 1.8346 \\ 9.5080 \\ 22.4026 \end{bmatrix}$$



Gambar 4.24 Hasil *crop* dari Matlab GUI untuk *Trajectory* Persegi

Hasil MSE dari masing-masing *error* yaitu *error leading*, *lateral*, dan orientasi didapatkan sebesar 3.3×10^{-3} m, 4.4×10^{-3} m, dan 5.3×10^{-3} radian. Sedangkan MSE dari *error* kecepatan linier dan angular *input auxiliary velocity* yaitu sebesar 2.96×10^{-3} m/s dan 0.1939 rad/sec. Perbandingan hasil dari pengujian dengan satu dan dua pengukuran diberikan pada Tabel 4.5.

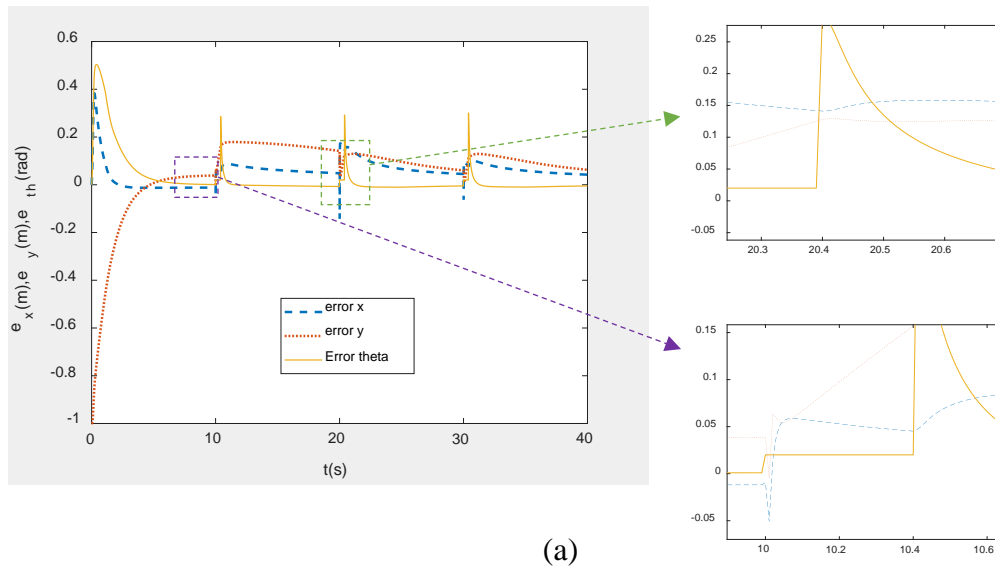
Nilai *settling time* yang dihasilkan dengan dua pengukuran (2.6,4.5,2.5) lebih baik dari satu pengukuran satu pengukuran (16,15,35) pada *trajectory* segi delapan. Demikian juga pada pengujian persegi dan segienam seperti yang ditunjukkan oleh Tabel 4.5 namun untuk nilai MSE *error lateral* pada pengujian dengan satu pengukuran lebih baik dibandingkan dua pengukuran pada semua *trajectory*. Nilai MSE yang lebih baik pada *error lateral* ini diakibatkan oleh pengaturan nilai variabel *premise*. Nilai *premise* e_y pada setiap pengujian dibuat besar karena tujuan tesis adalah meminumkan *error* dan memperoleh *settling time* yang cepat (untuk membedakan dengan kontrol pada [11]) sehingga *overshoot error lateral* pada awal pergerakan membuat nilai MSE menjadi besar.

Tabel 4.5 Perbandingan *Trajectory Tracking* dengan Satu Pengukuran dan Dua Pengukuran

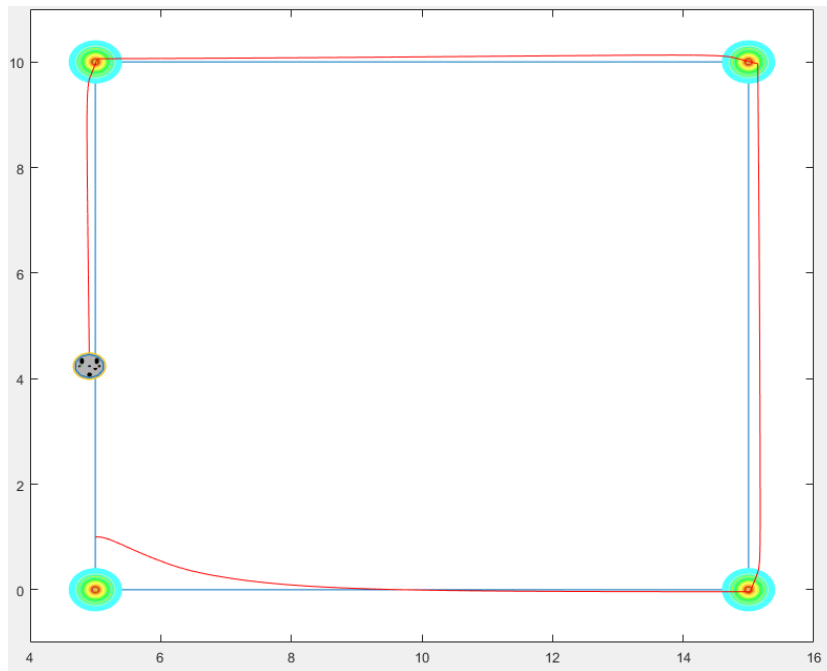
Trajectory	Error	Satu Pengukuran		Dua Pengukuran	
		<i>Settling time</i>	MSE	<i>Settling time</i>	MSE
Garis Lurus	e_x	0.9438	1.6×10^{-3}	-	-
	e_y	2.18	4.1×10^{-2}	-	-
	e_θ	2.35	1.3×10^{-2}	-	-
Segi Delapan	e_x	16	4.3×10^{-3}	2.6	7.9×10^{-4}
	e_y	15	1.4×10^{-3}	4.5	1.5×10^{-3}
	e_θ	35	2×10^{-3}	2.5	1.2×10^{-3}
Persegi	e_x	3.43	4.6×10^{-3}	2.1	5.4×10^{-3}
	e_y	4.2	1.82×10^{-2}	4.5	2.2×10^{-3}
	e_θ	30.4	1.8×10^{-3}	4.4	6.2×10^{-3}
Segi Enam	e_x	5	2.8×10^{-3}	2	4.7×10^{-3}
	e_y	10	5.2×10^{-3}	4.2	6.4×10^{-3}
	e_θ	36	8.4×10^{-3}	11.7	1.3×10^{-3}

Selanjutnya dilakukan pengujian untuk memperoleh data jarak antar magnet yang dapat diatur sehingga AGV tetap dapat melakukan *trajectory tracking* dengan *error* yang minimum yaitu ± 0.1 m untuk e_x dan e_y sedangkan *error* orientasi ± 0.1 radian. Jarak magnet yang digunakan pada pengujian sebelumnya adalah 5 m selanjutnya akan diuji penggunaan jarak sebesar 10 m.

Untuk hasil pengujian ditunjukkan pada Gambar 4.25 dimana dengan jarak tersebut AGV tidak dapat melakukan *tracking* dengan baik. Hal ini dikarenakan *error* orientasi mengalami *overshoot* sebesar 300% yang mengakibatkan tidak didapatkannya hasil koreksi menggunakan pengukuran MFD walaupun didapatkan konvergensi *error* setelah AGV berbelok. Dengan demikian jarak maksimal yang dapat digunakan agar AGV dapat melakukan *trajectory tracking* dengan *error* yang minimum yaitu 5 m untuk *trajectory* persegi dengan kecepatan referensi sebesar 1 m/s.



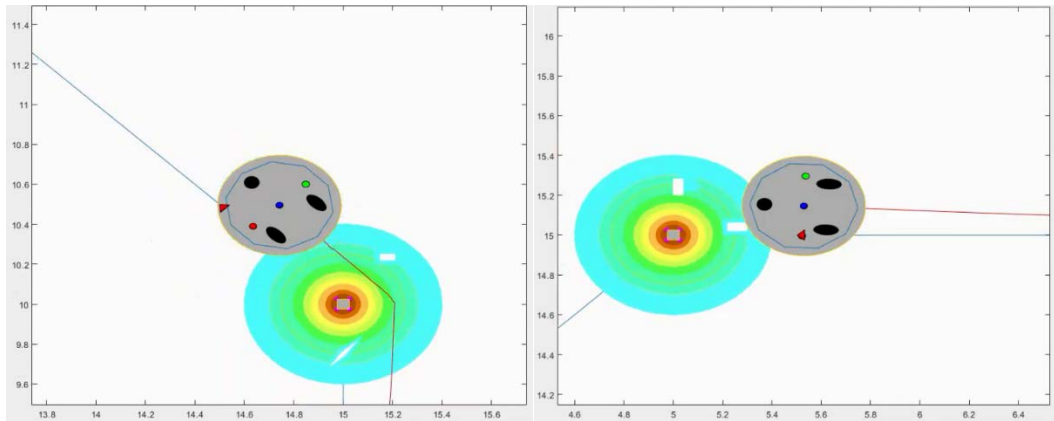
(a)



(b)

Gambar 4.25 Pengujian Jarak antar *Magnetic Spot* (a) *Error Tracking* (b) Hasil *Tracking*

Untuk pengujian *trajectory tracking* dengan gangguan eksternal ditunjukkan pada Tabel 4.6. Pada pengujian dengan disturbance didapatkan hasil yang cukup baik walaupun nilai *error in leading* dan *error lateral* mengalami kenaikan yang kecil terutama pada *trajectory* persegi. Hal ini terjadi karena *error tracking* yang terjadi sebelum memasuki area MFD cukup besar sehingga hasil koreksi cukup kecil.



Gambar 4.26 Hasil *Tracking* dengan Gangguan Eksternal

Tabel 4.6 Hasil *Trajectory Tracking* menggunakan Dua Pengukuran dengan Gangguan Eksternal

Trajectory	$Error$	Dua pengukuran dengan gangguan	
		$Settling\ time$	MSE
Segi Delapan	e_x	3.6	7.31×10^{-4}
	e_y	4.5	1.6×10^{-3}
	e_θ	7.2	1.2×10^{-3}
Persegi	e_x	4.8	5.4×10^{-3}
	e_y	6.2	2.2×10^{-3}
	e_θ	7.4	6.2×10^{-2}
Segi Enam	e_x	6.9	4.7×10^{-3}
	e_y	9.2	6.4×10^{-3}
	e_θ	13.7	1.3×10^{-3}

Dari ketiga pengujian didapatkan nilai *overshoot error* orientasi yang minimum sehingga dapat disimpulkan kontroler mampu mengatasi gangguan kecil pada pengukuran MFD.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat ditunjukkan bahwa untuk permasalahan *tracking* dari AGV pada *trajectory* dengan perubahan sudut yang besar dan pemberian inisial *error* dapat diselesaikan menggunakan kontroler fuzzy Takagi-Sugeno sebagai penghasil *input auxiliary velocity* untuk kontroler dinamika. Untuk hasil dari penggunaan satu pengukuran dan model kinematika (3.2) didapatkan nilai *steady state error tracking* dibawah 0.01 m dan 0.01 radian pada semua *trajectory*. Namun permasalahan yang terjadi ketika menggunakan satu pengukuran dan model dinamika (3.15) adalah besarnya *error* orientasi (e_θ) disetiap belokan pada *trajectory* segidelapan, persegi, dan segienam yaitu sebesar 0.85, 2.26, dan 2.26 radian. Walaupun konvergensi *error* tercapai, *settling time* dari e_θ yang dihasilkan juga cukup besar yaitu lebih dari 30 s pada semua *trajectory* akibat adanya *overshoot* dengan nilai lebih dari 300% pada setiap belokan.

Untuk penggunaan dua pengukuran sebagai *feedback* dimana pengukuran kedua digunakan sebagai koreksi *error* pergerakan AGV dari pengukuran pertama didapatkan nilai *overshoot* yang minimum. Nilai *overshoot* yang terjadi disetiap belokan dengan nilai tidak lebih dari 50% pada semua *trajectory* baik untuk e_x , e_y , dan e_θ . Untuk nilai *steady error* yang dihasilkan yaitu $e_x < 0.05$ m, $e_y < 0.1$ m, dan $e_\theta < 0.1$ radian. Kemudian untuk nilai minimum *settling time* yang dihasilkan pada satu pengukuran menggunakan model kinematika lebih baik dari metode sebelumnya. Untuk hasil rancangan rata-rata nilai *settling time* dari e_x , e_y , dan e_θ adalah 7.5 s, 5 s, dan 5.6 s sedangkan hasil metode sebelumnya yaitu 15 s, 17, s, dan 11 s. Untuk dua pengukuran menggunakan model dinamika nilai *settling time* terbesar adalah 12 s untuk semua *error*.

Dengan demikian dapat disimpulkan bahwa sistem kontrol hasil rancangan dapat meminimumkan *error tracking* pada *trajectory* dengan perubahan sudut yang besar dan juga dihasilkan *settling time* yang minimum dibandingkan dengan metode sebelumnya. Sistem kontrol juga mampu mengatasi adanya gangguan kecil pada MFD berupa besi dengan dihasilkannya *steady state error* dibawah 0.1 m.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] H. Fazlollahtabar and M. S. Mehrabad, *Autonomous Guided Vehicles*, vol. 45, Cham, Switzerland: Springer, 2015, pp. 135-156.
- [2] G. Ullrich, *Automated Guided Vehicle*, Heidelberg, Germany: Springer-Verlag, 2015.
- [3] H. W. Yang and S. Y. Lee, "Navigation of Automated Guided Vehicles Using Magnet Spot Guidance Method," *Robotic and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 425-436, 2012.
- [4] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A Stable Tracking Control Method for a Non-Holonomic Mobile Robot," *Intelligent Robots and Systems IROS'91*, vol. 3, pp. 1236-1241, 1991.
- [5] E. H. Guechi, J. Lauber, M. Dambrine, G. Klancar and S. Blazic, "PDC Control Design for Non-holonomic Wheeled Mobile Robot with Delayed Output," *J Intell. Robot Syst.*, pp. 395-414, 2010.
- [6] H. Yang, X. Fan and Y. Xia, "Robust Tracking Control for Wheeled Mobile Robot Based on Extended State Observer," *Advanced Robotic*, vol. 30, no. 1, pp. 1-10, 2015.
- [7] R. Fierro and F. L. Lewis, "Control of a Nonholomic Mobile Robot: Backstepping Kinematics Into Dynamics," *IEEE Conference on Decision and Control*, vol. 4, pp. 3805-3810, 1995.
- [8] R. Dhaouadi and A. A. Hatab, "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: a Unified Framework," *Advances in Robotics & Automation*, vol. 2, no. 2, pp. 1-7, 2013.
- [9] H. M. Shen, L. Hu, H. L. Qin and X. Fu, "Real-Time Orientation-Invariant Magnetic Localization and Sensor Calibration Based on Closed-Form Models," *IEEE Magnetic Letters*, vol. 6, 2015.
- [10] Y. Pang, A. L. D. L. Cruz and G. Lodewijks, "Bipolar Magnetic Positioning System For Automated Guided Vehicle," *IEEE Intelligent Vehicles Symposium*, pp. 883-888, 2008.
- [11] S. Blazic and M. Bernal, "Trajectory Tracking for Nonholomic Mobile Robot Based on Extended Models," *International Federation of Automatic Control*, pp. 5938-5943, 2011.

- [12] J. H. Lee and S. Jung, "Global Position Tracking Control of an Omni-Directional Mobile Robot using Fusion of a Magnetic Compass and Encoders," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 246-251, 2008.
- [13] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, New York, NY, USA: Jhon Wiley & Sons Inc, 2001.
- [14] G. Klancar, A. Zdesar, S. Blazic and I. Skrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*, Butterworth-Heinemann, 2016.
- [15] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Pretince-Hall, 1991.
- [16] A. Kelly, *Mobile Robotics : Mathematics, Models, and Methods*, Cambridge University Press, 2013.
- [17] K. Ogata, *Modern Control Engineering*, 5th ed., Pearson, 2011.
- [18] M. Vidyasagar, *Nonlinear Systems Analysis*, Englewood Cliffs: Prentice Hall Inc, 1978.
- [19] T. Agustinah, J. Achmad and N. Mohammad, "Fuzzy Tracking Control Based on H_{∞} Performance for Nonlinear Systems," *WSEAS Trans. on Systems and Control*, vol. 6, no. 11, pp. 393-403, 2011.
- [20] M. S. Masmoudi and N. krichen, "Fuzzy Logic Controllers Design For Omnidirectionnal Mobile Robot Navigation," *Applied Soft. Computing*, vol. 49, p. 901919, 2016.
- [21] X. Li and C. Luo, "A Fuzzy PID Controller Applied in AGV Control System," *International Conference on Advanced Robotic and Mechatronic*, pp. 555-560, 2016.
- [22] R. Fierro and F. Lewis, "Control of a Nonholomic Mobile Robot Using Neural Networks," *IEEE trans. Neural Networks*, vol. 9, pp. 589-600, 1998.
- [23] Q. Chen, Y. Gu and F. Ding, "Data Filtering Based Recursive Least Squares Estimation Algorithm For a Class of Wiener Nonlinear Systems," *Intelligent Control and Automation*, pp. 1848-1852, 2014.
- [24] T. Braumi, "Mobile Robot Design and Applications with Embedded Systems," in *Embedded Robotics*, Berlin Heidelberg, Springer, 2006, pp. 9-98.

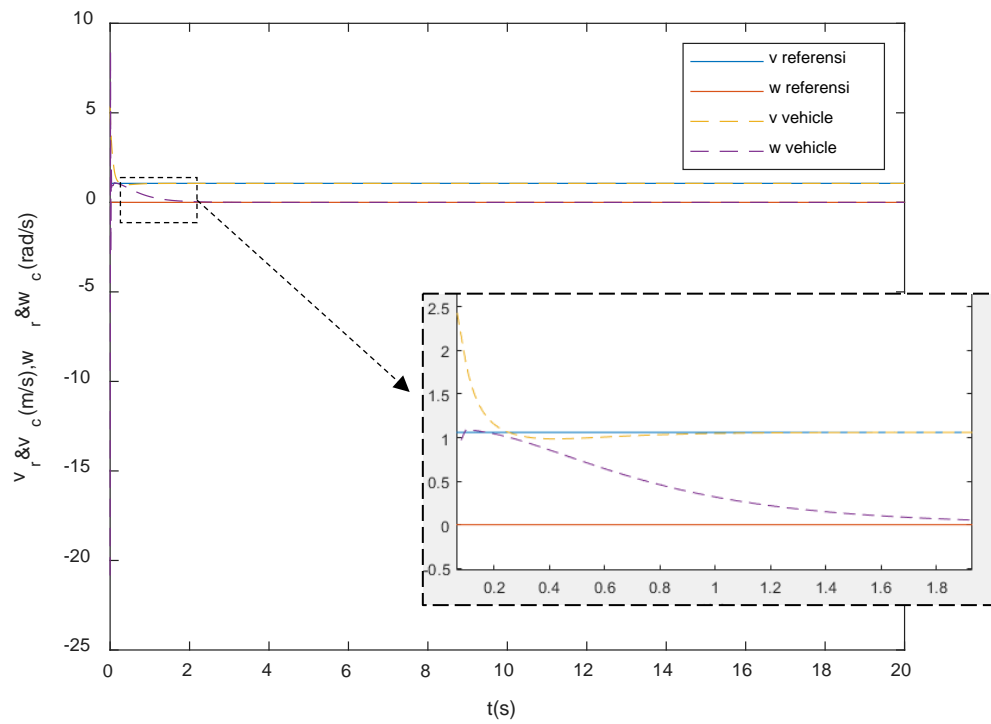
- [25] J. Tan, K. Liu and G. Turk, "Stable Proportional-Derivative Controllers," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 34-44, 2011.
- [26] B. S. Cole, "Magnetic Field-Based Navigation of a Mobile Robot," Oklahoma State University, Oklahoma, 2005.
- [27] E. P. Furlani, Permanent Magnet and Electromechanical Devices: Materials, Analysis, and Applications, san diego, CA, USA: Academic, 2001.
- [28] Z. Song, X. Wu, T. Xu, J. Sun, Q. Gao and Y. He, "A New Method of AGV Navigation Based on Kalman Filter and a Magnetic Nail Localization," *International Conference on Robotics and Biomimetics*, pp. 952-957, 3-7 Dec 2016.
- [29] C. M. Lin and Y. J. Mon, "A Fuzzy-PDC Based Control for Robotic Systems," *Information Sciences*, vol. 137, pp. 135-155, 2001.
- [30] S. Kamekawa and S. Uemura, "A Magnetic Guidance Method for Automated Guided Vehicles," *IEEE Transactions On Magnetic*, vol. 23, no. 5, pp. 2416-2418, 1987.

Halaman ini sengaja dikosongkan

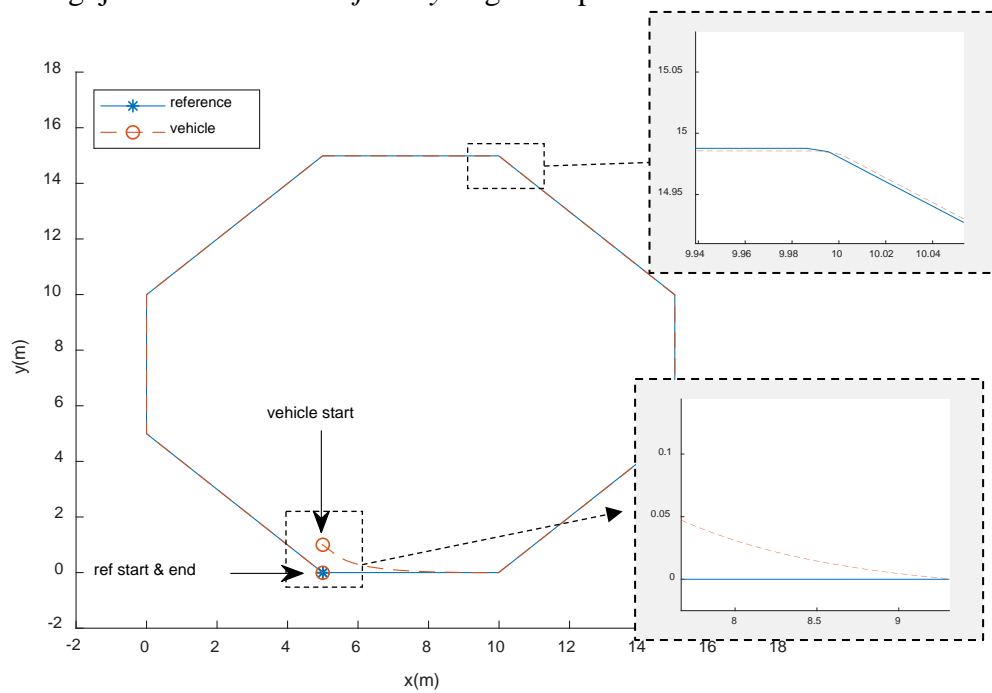
LAMPIRAN

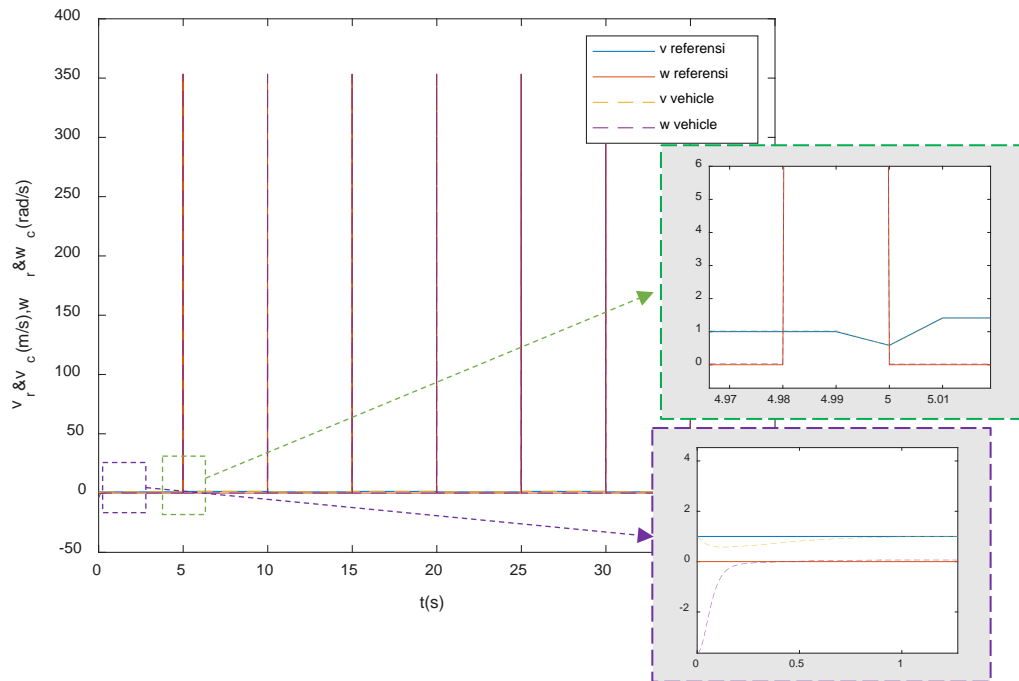
LAMPIRAN A

A.1 Hasil untuk Pengujian 4.2.1 pada Garis Lurus

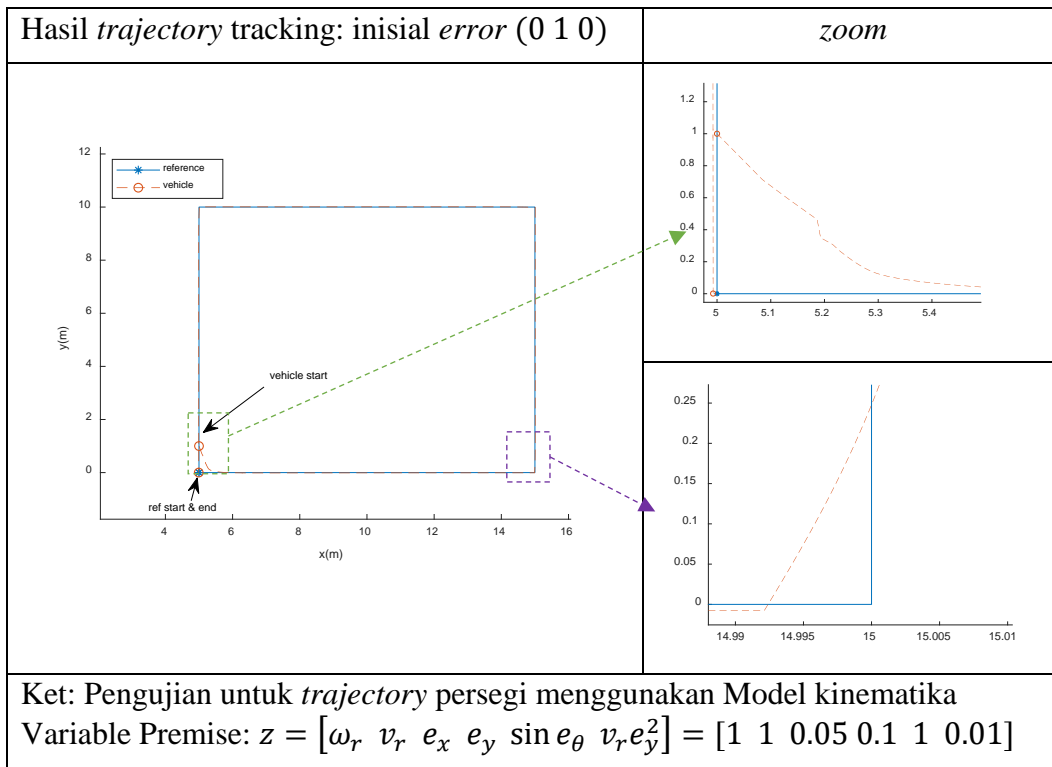


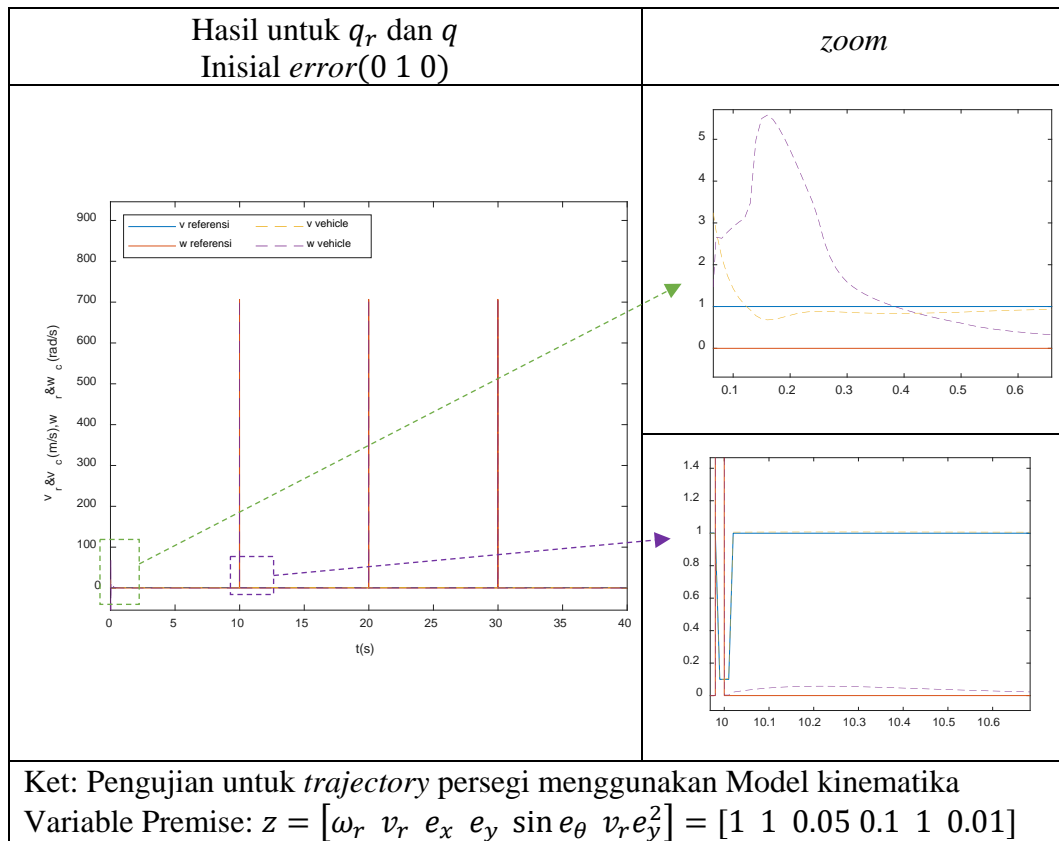
A.2 Pengujian 4.2.1 untuk *Trajectory* Segi Delapan



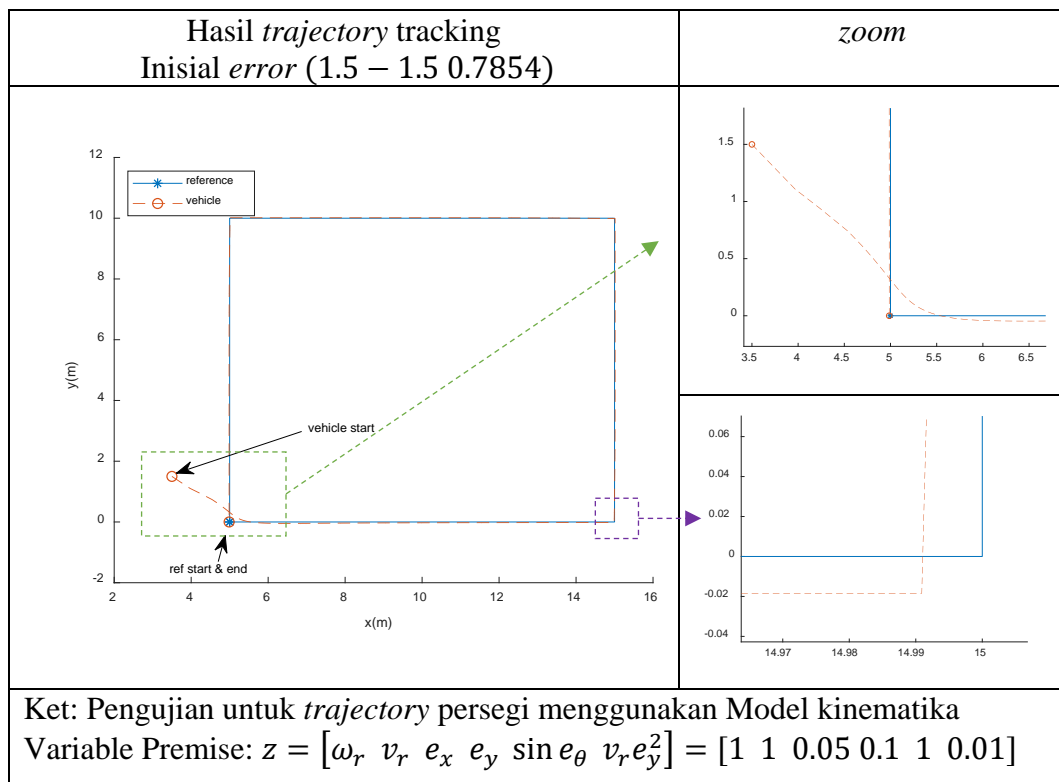


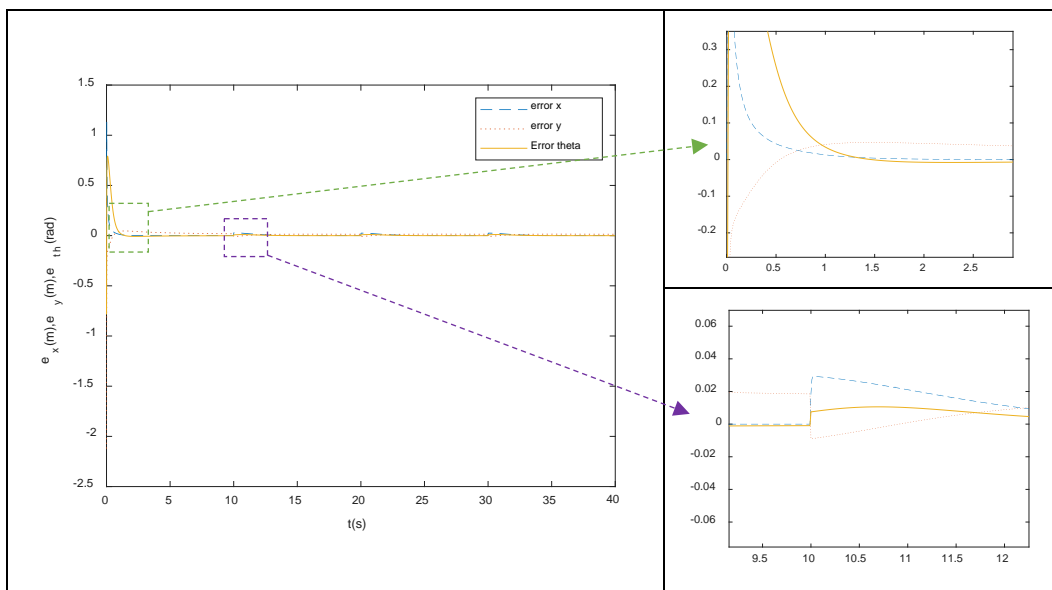
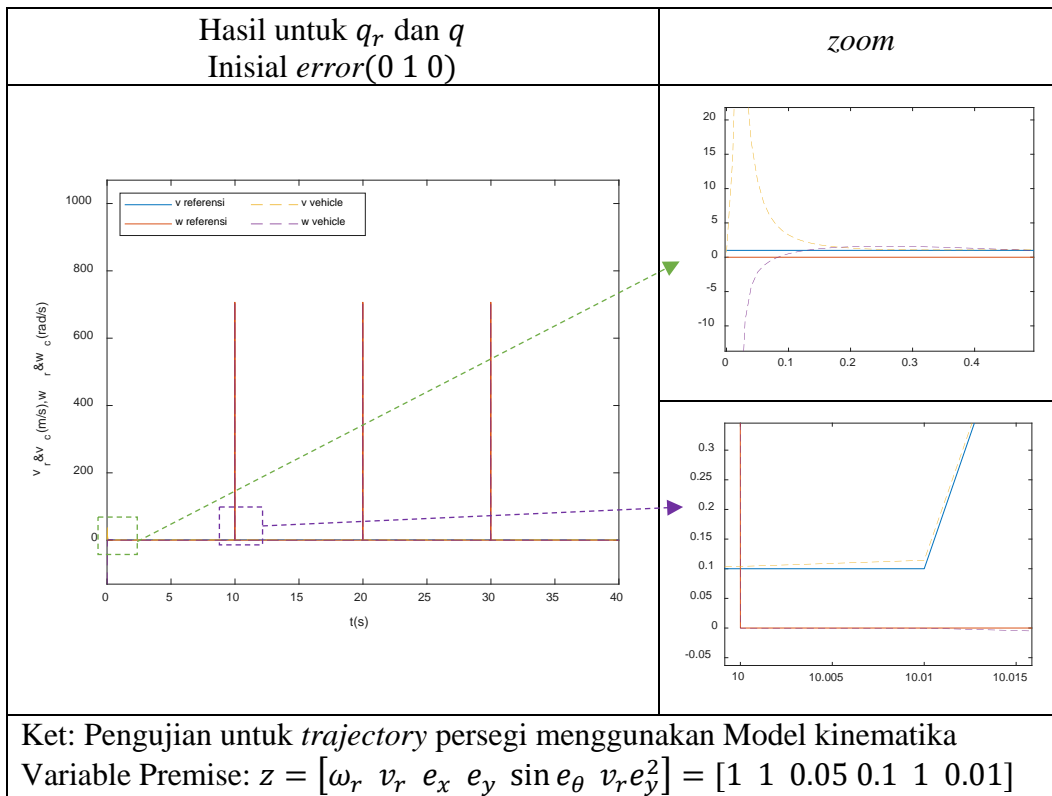
A.3 Pengujian 4.2.1 untuk *Trajectory* Persegi





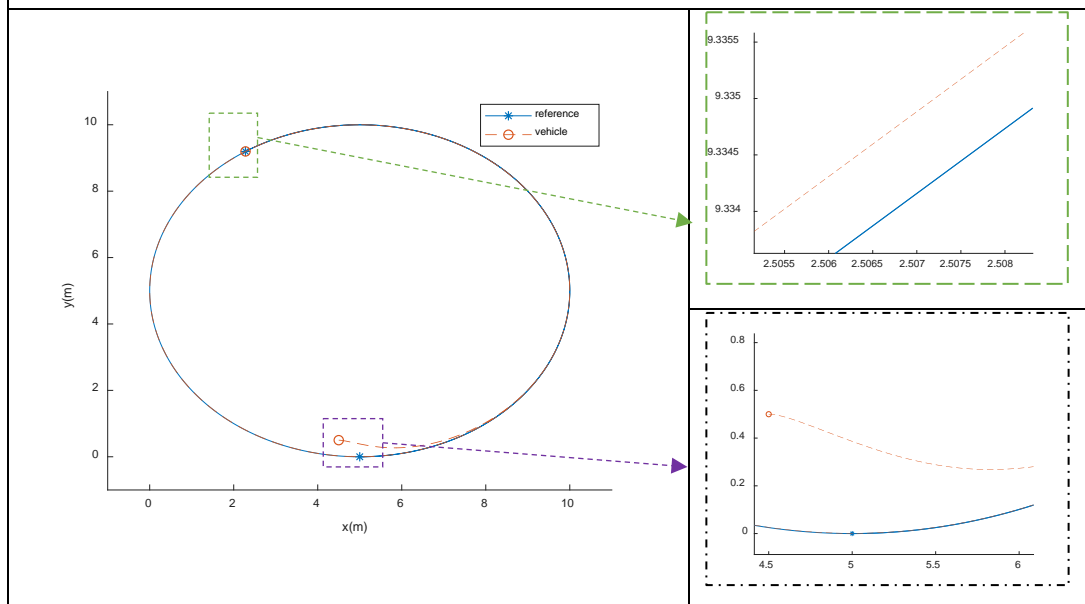
A.4 Pengujian 4.2.1 untuk *Trajectory* Persegi dengan Inisial *error* berbeda



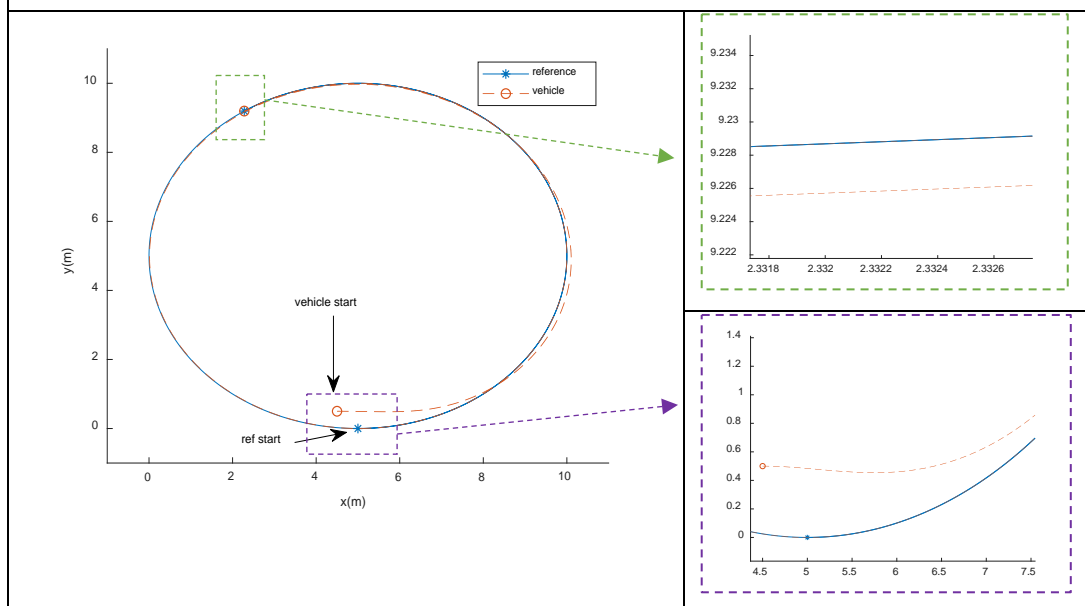


A.5 Hasil Pengujian 1 sub-bab 4.2.2 Tabel 4.2

Pengujian Hasil Perancangan Kontrol menggunakan model kinematika

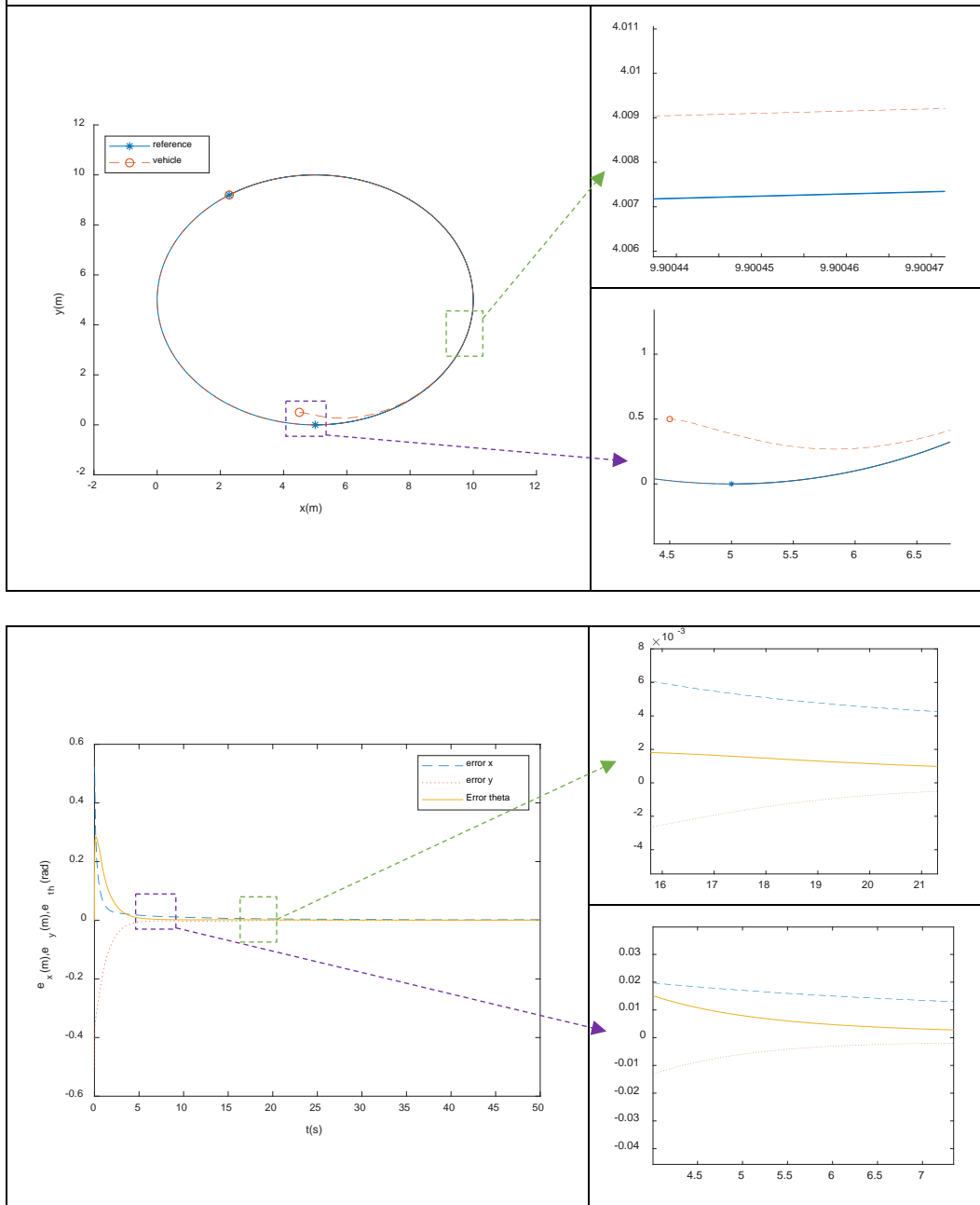


Pengujian dengan kontroler [11] menggunakan model kinematika

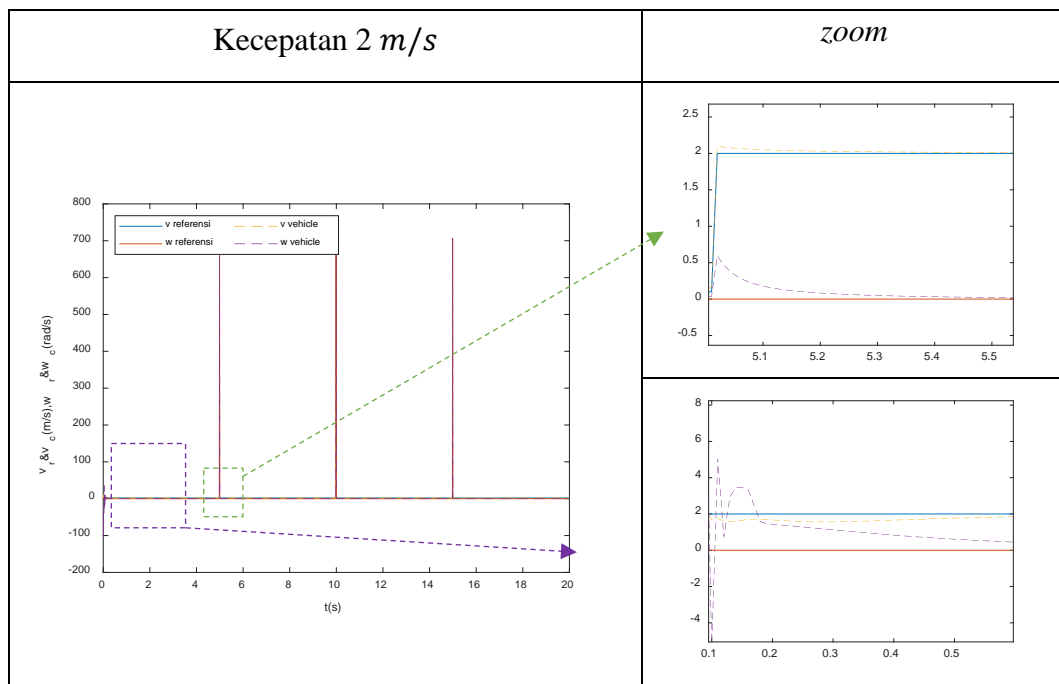
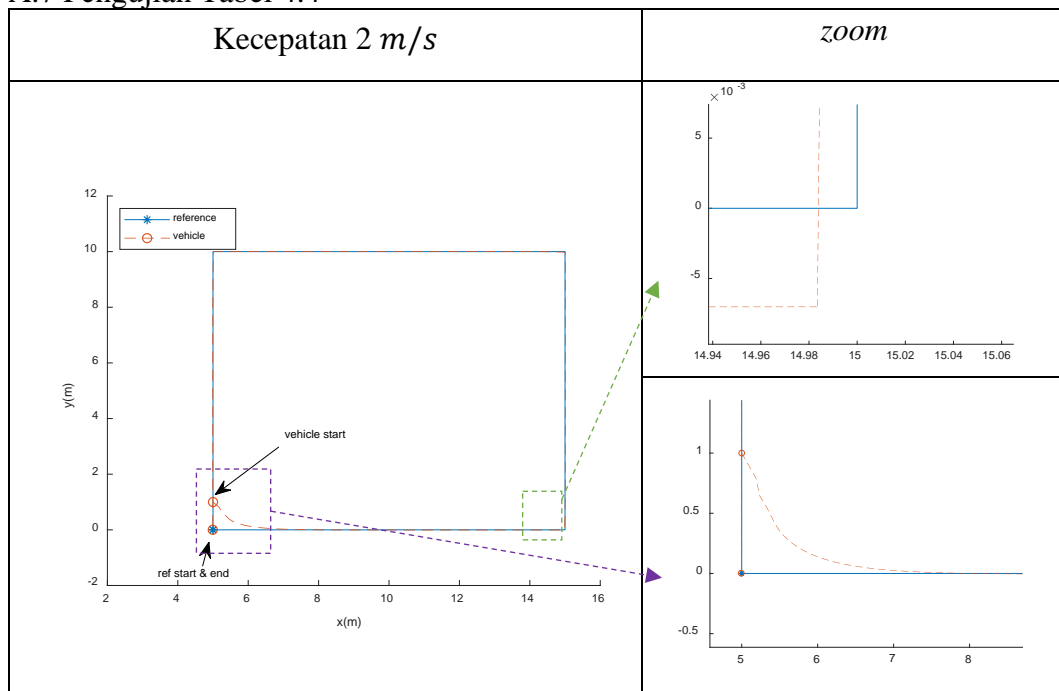


A.6 Hasil Pengujian 2 sub-bab 4.2.2 Tabel 4.2

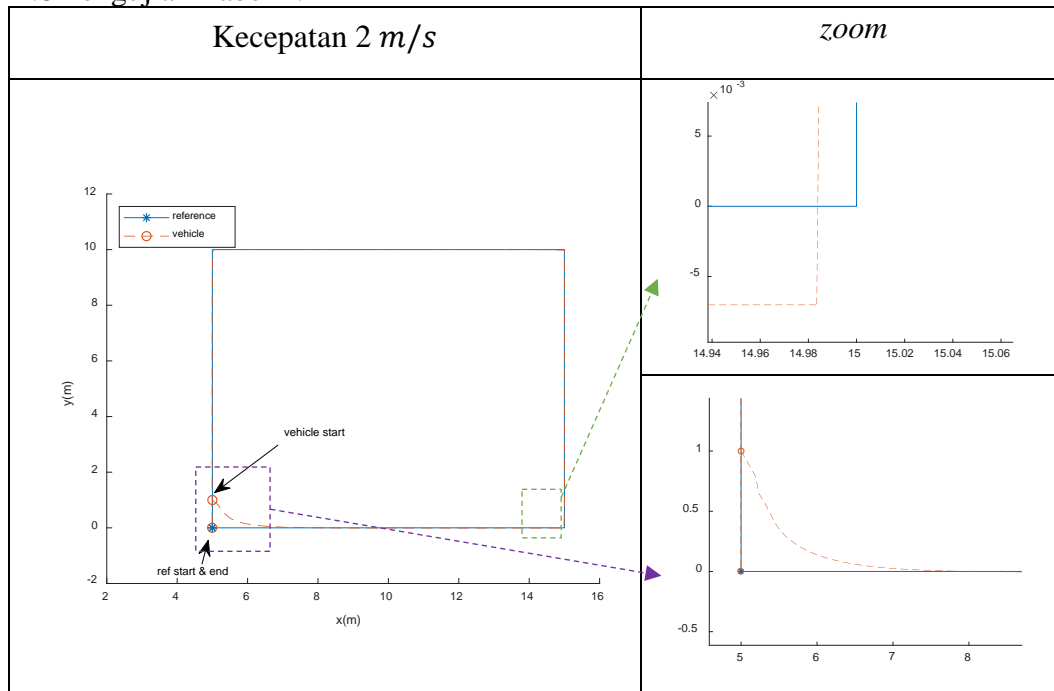
Pengujian hasil perancangan kontroler



A.7 Pengujian Tabel 4.4



A.8 Pengujian Tabel 4.4



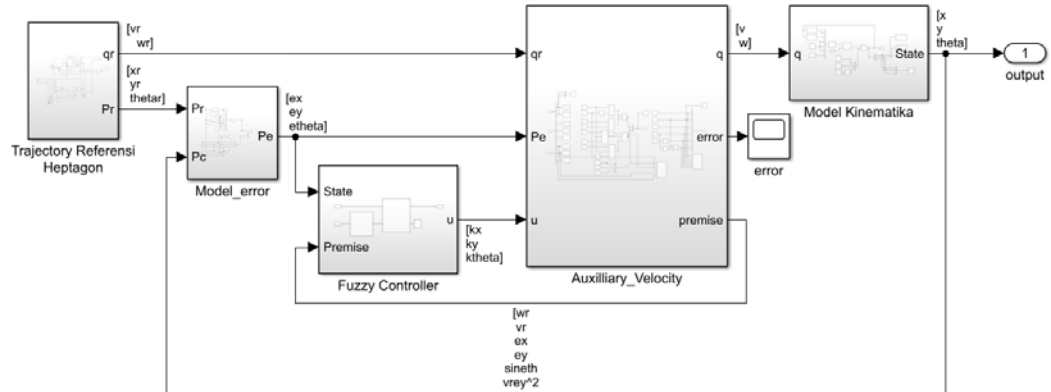
A.9 Hasil Matrik Stabilitas P dan Q dari pengujian pada Tabel 4.4 dan 4.5

Kecepatan (m/s)	Matriks P	Matriks Q
1.5	$\begin{bmatrix} 5.9230 & 13.0573 & 2.3886 \\ 13.0573 & 50.2178 & 9.2048 \\ 2.3886 & 9.2048 & 28.0616 \end{bmatrix}$	$\begin{bmatrix} 1.0432 & -0.2385 & -0.0008 \\ -0.2385 & 0.1921 & -0.0328 \\ -0.0008 & -0.0328 & 0.0652 \end{bmatrix}$
3	$\begin{bmatrix} 5.9230 & 13.0573 & 2.3886 \\ 13.0573 & 50.2178 & 9.2048 \\ 2.3886 & 9.2048 & 28.0616 \end{bmatrix}$	$\begin{bmatrix} 1.0432 & -0.2385 & -0.0008 \\ -0.2385 & 0.1921 & -0.0328 \\ -0.0008 & -0.0328 & 0.0652 \end{bmatrix}$
5	$\begin{bmatrix} 2.3939 & 2.4439 & 4.4061 \\ 2.4439 & 24.1923 & 23.3770 \\ 4.4061 & 23.3770 & 61.0905 \end{bmatrix}$	$\begin{bmatrix} 0.0031 & 0.0000 & -0.0001 \\ 0.0000 & 0.0028 & -0.0010 \\ -0.0001 & -0.0010 & 0.0004 \end{bmatrix}$

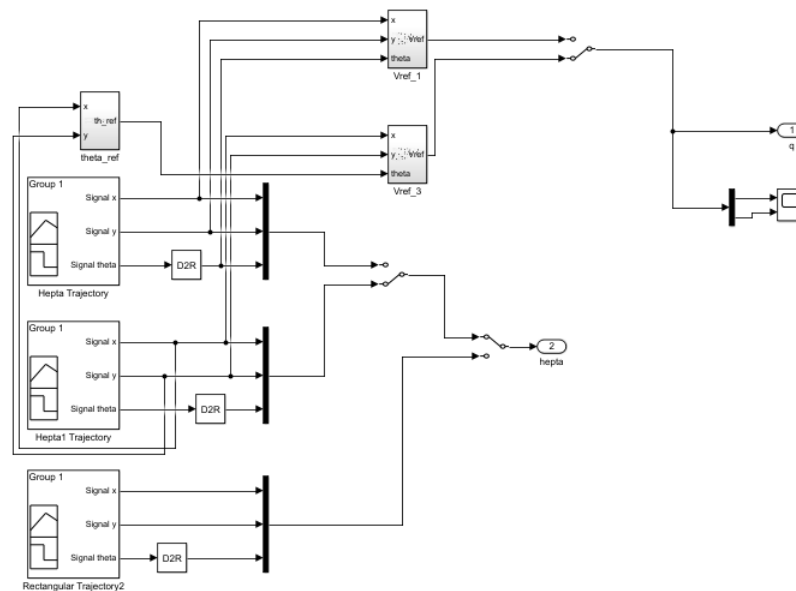
Kecepatan (m/s)	Matriks P	Matriks Q
1	$\begin{bmatrix} 0.8288 & 0.0008 & 0.0002 \\ 0.0008 & 10.7194 & 1.6500 \\ 0.0002 & 1.6500 & 1.0463 \end{bmatrix}$	$\begin{bmatrix} 0.1407 & -0.0000 & 0.0000 \\ -0.0000 & 0.1330 & -0.0385 \\ 0.0000 & -0.0385 & 0.1256 \end{bmatrix}$
2	$\begin{bmatrix} 0.8146 & 0.0023 & 0.0006 \\ 0.0023 & 15.4300 & 2.7009 \\ 0.0006 & 2.7009 & 1.2517 \end{bmatrix}$	$\begin{bmatrix} 0.0554 & -0.0000 & 0.0000 \\ -0.0000 & 0.0412 & -0.0097 \\ 0.0000 & -0.0097 & 0.0176 \end{bmatrix}$
4	$\begin{bmatrix} 0.3278 & 0.0040 & 0.0013 \\ 0.0040 & 9.5201 & 1.3053 \\ 0.0013 & 1.3053 & 0.4683 \end{bmatrix}$	$\begin{bmatrix} 0.2627 & -0.0001 & 0.0003 \\ -0.0001 & 0.0483 & -0.0085 \\ 0.0003 & -0.0085 & 0.0165 \end{bmatrix}$

Lampiran B

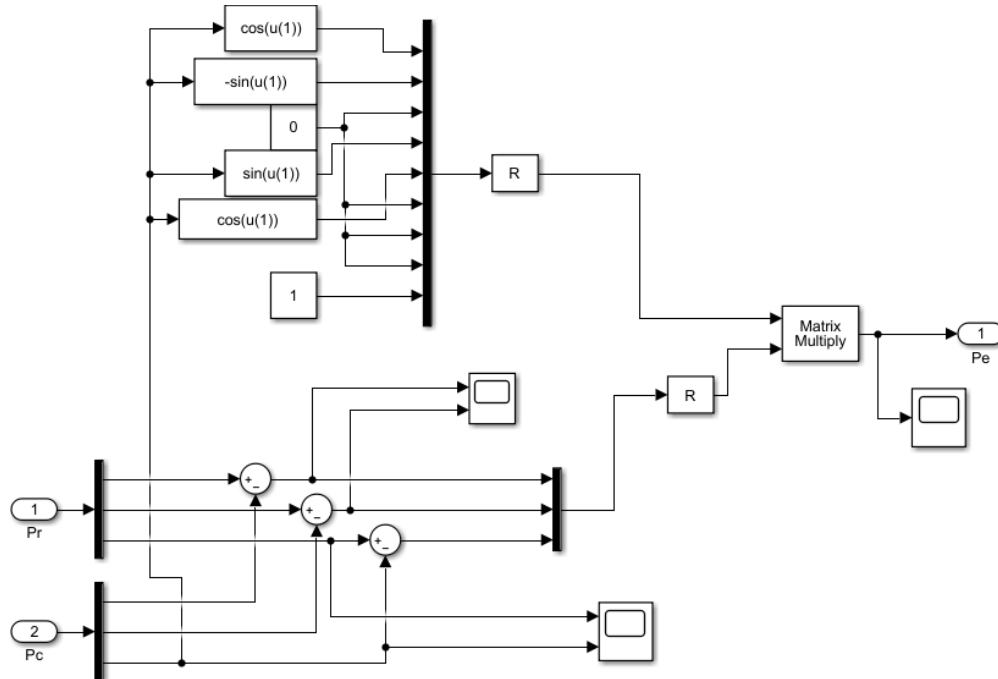
B.1. Blok Simulink untuk pengujian menggunakan *plant* kinematika



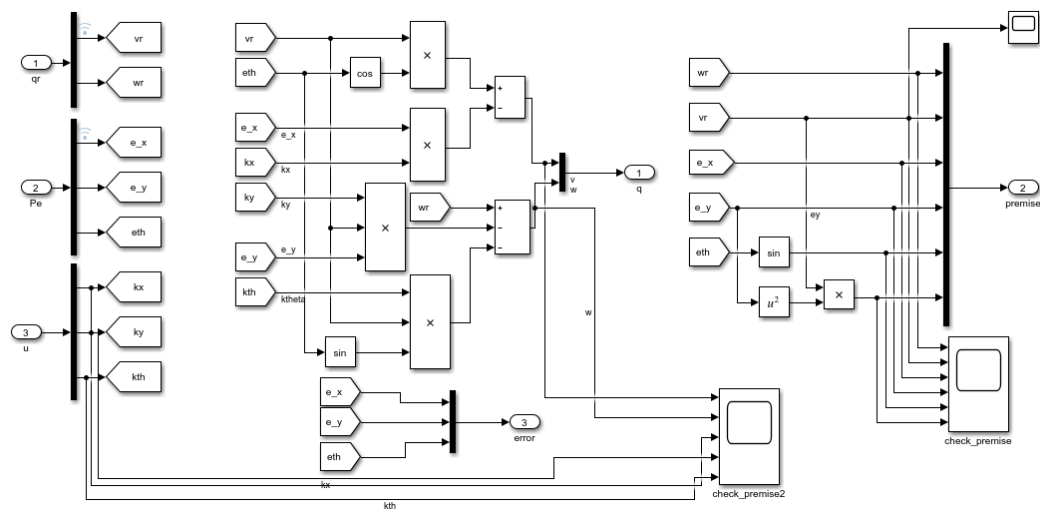
B.2 Blok *Trajectory* referensi



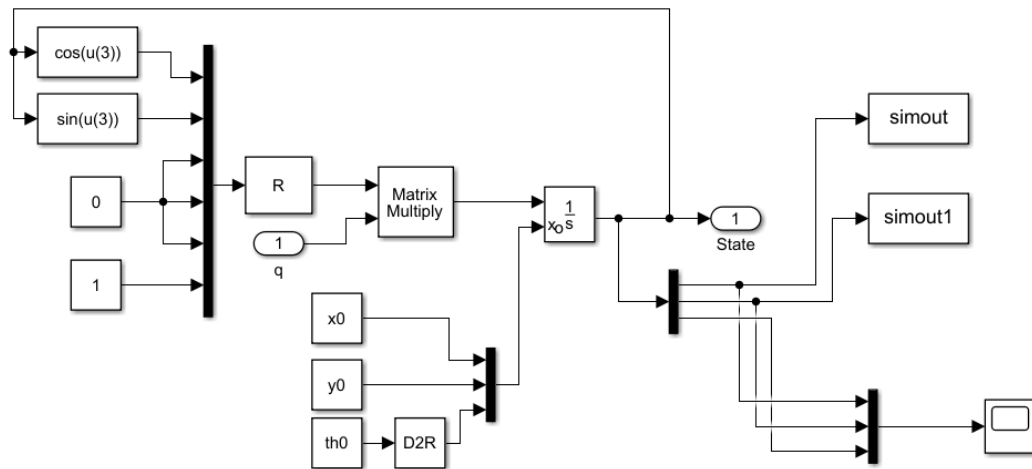
B.3 Blok Model Error



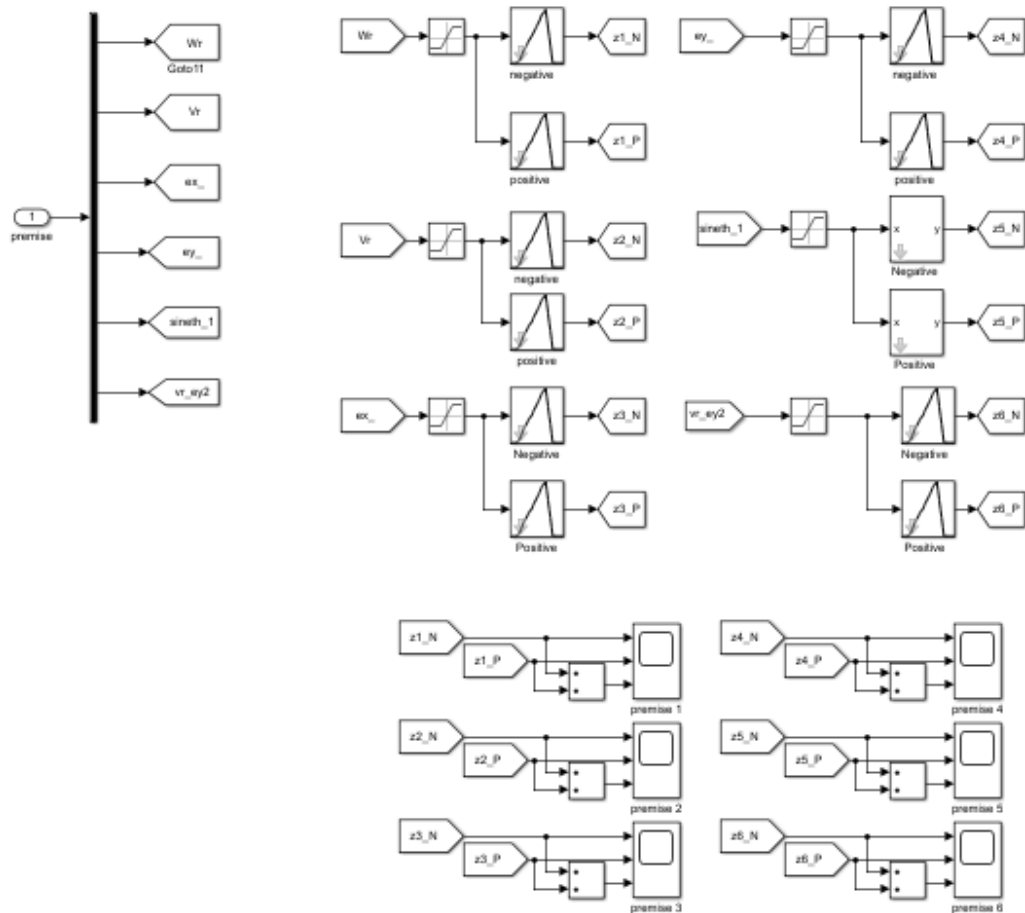
B.4 Blok Auxiliary velocity



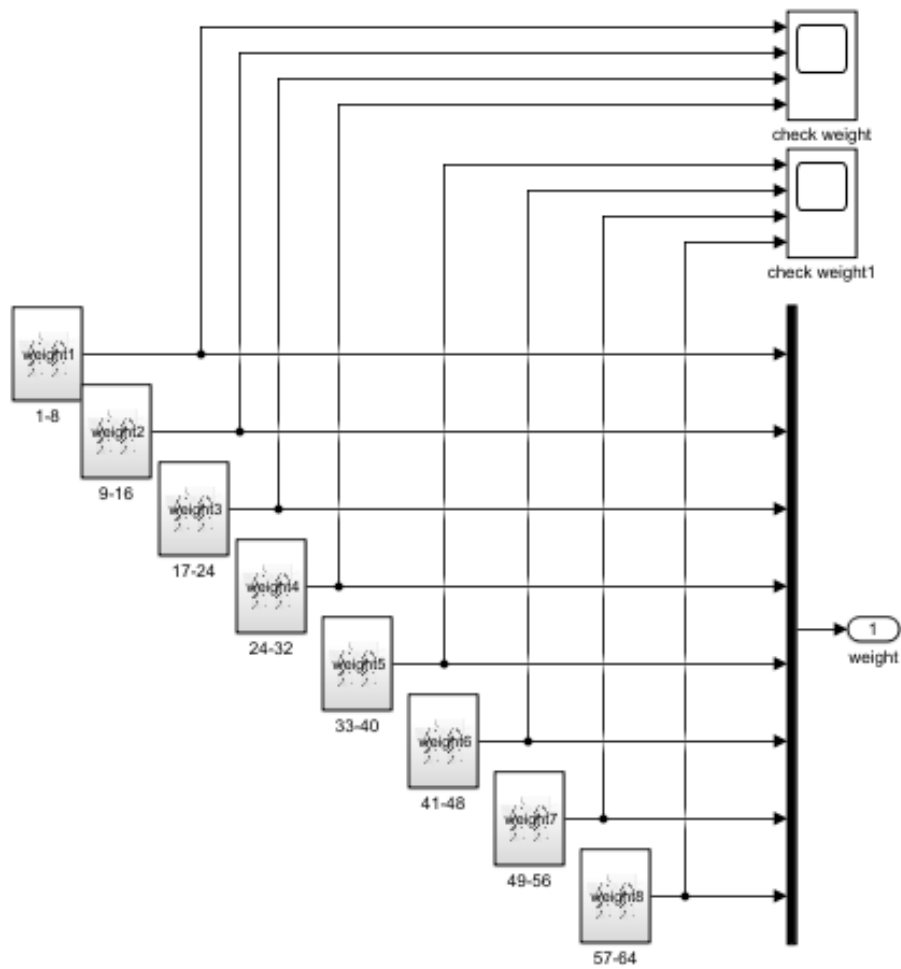
B.5 Model Kinematika berdasarkan kecepatan linear dan angular



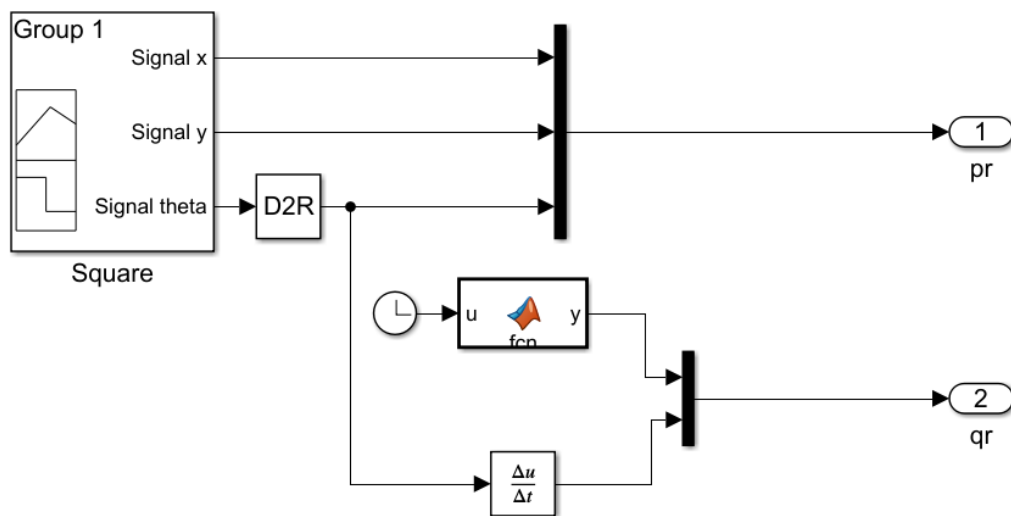
B.6 Membership function



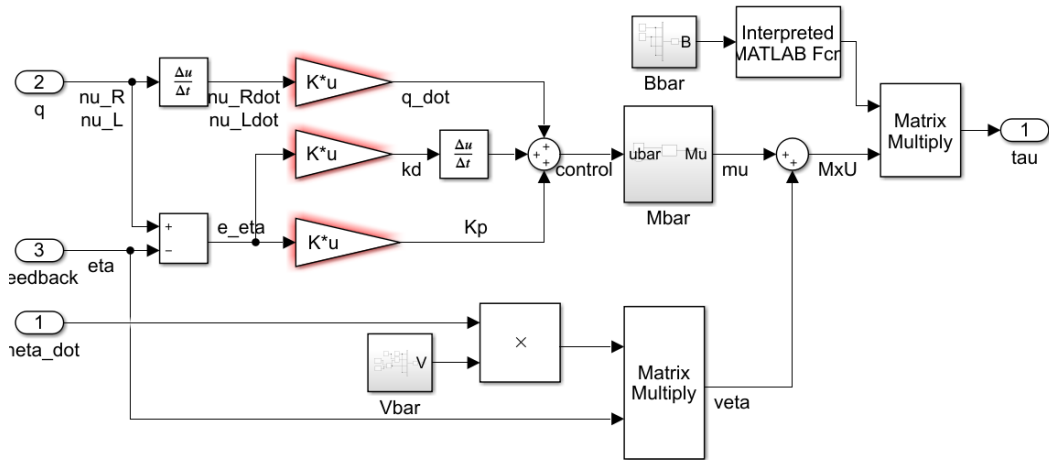
B.7 Pengaturan Bobot



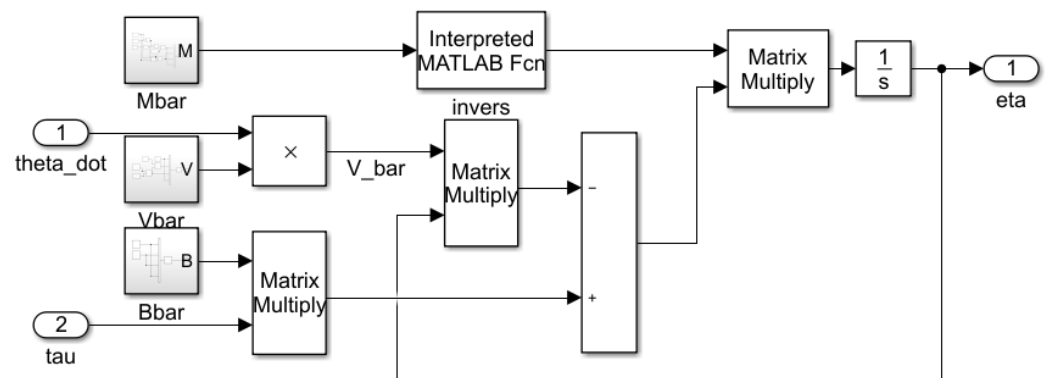
B.8 Simulink *Trajectory* referensi



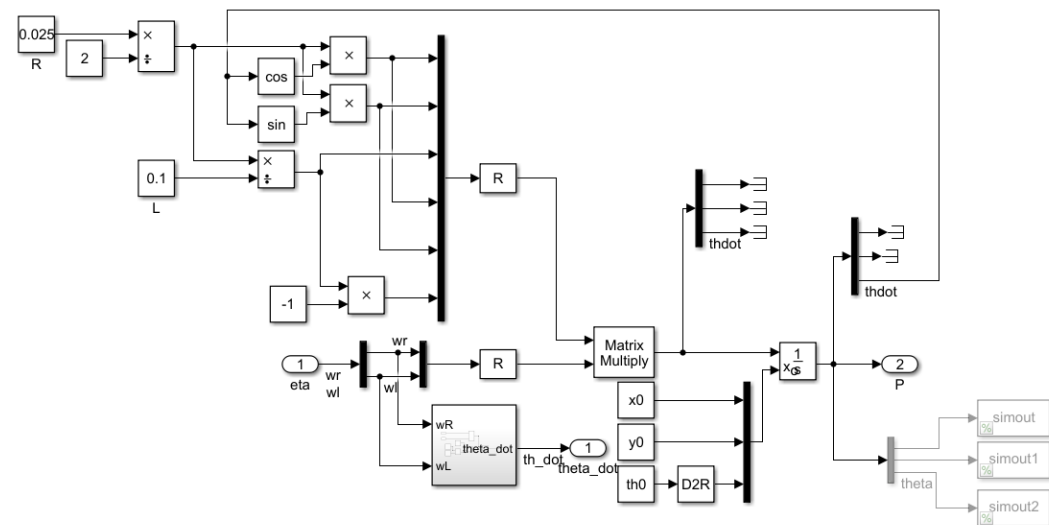
B.9 Simulink Kontroler Dinamika



B.10 Blok Model Dinamika



B.11 Blok Model Kinematika



Halaman ini sengaja dikosongkan

Lampiran C

C.1 Pengecekan rank sistem (2.14)

```
%Pengecekan dilakukan untuk nilai theta 0rad-360rad
ey=0; %titik uji (equilibrium)
ex=0; %titik uji
for th=0:360      %errorsudut(theta)
    tth=-th;
    for s=0:200    %iterasi
        wr=s/100;  %kecepatan angular wr
        A=[0 wr 0;-wr 0 r*sin(-th);0 0 0];%matriks A
        B=[-1 ey;0 -ex;0 -1];           %matriks B
        k=rank(A);                       %cek rank matriks A
        kc=rank(B);                      %cek rank matriks B
        co=ctrb(A,B);                    %matriks controllability
        ksis(th+1,s+1)=rank(co);         %hitung rank matriks
                                           %controllability

        unco=length(A)-rank(co);
    end
end
```

C.2 Pengecekan rank sistem (3.8)

```
%preallocation memory
ksis=zeros(361,201);
%pemilihan nilai parameter
ey=0; %titik uji (variabel)
ex=0; %titik uji (variabel)
for th=0:360      %errorsudut(theta)
    tth=-th;
    for s=0:200    %iterasi
        vr=s/10;   %kecepatan linear (variabel)
        wr=-s/100; %kecepatan angular (variabel)
        A=[0 wr 0;-wr 0 wr*sin(-th);0 0 0];%matriks A
        B=[-ex      vr*ey.^2      vr*ey*sin(th)
            0        -vr*ex*ey      -vr*ex*sin(th)
            0        -vr*ey         -vr*sin(th)]; %matriks B
        k=rank(A);                       %cek rank matriks A
        kc=rank(B);                      %cek rank matriks B
        co=ctrb(A,B);                    %matriks controllability
        ksis(th+1,s+1)=rank(co);         %hitung rank matriks
                                           %controllability

        unco=length(A)-rank(co);
    end
end
```

C.3 Magnetik field dari (3.18)

```
%parameter
Br=13400;
L=0.1; %long
T=0.2; %0.25 thick
W=0.1; %wide
sum=zeros(1,40);
sum1=zeros(1,40);
```

```

g=zeros(1,5);
M=1.7e6;
for S=1:1:40
    X=S/100;
    g(1)=atan((L*T)/2*(X+W)*(sqrt(L^2+T^2+4*(W+X)^2)));
    g(2)=atan((L*T)/2*X*sqrt(T.^2+L.^2+4*(X.^2)));
    g(3)=1.25663706e-6;
    Bx=(Br/pi)*(g(2)-g(1));
    sum(1,S)=Bx;
end
for S1=1:1:40
    syms cari
    Bx=sum(1,S1);
    eqn=Bx==(Br/pi)*(atan((L*T)/2*cari*sqrt(T^2+L^2+4*(cari^2)))-
    atan((L*T)/2*(cari+W)*(sqrt(L^2+T^2+4*(W+cari)^2))));
    solx=vpasolve(eqn,cari);
    sum1(1,S1)=solx;
end

```

C.4 Pengukuran koordinat dari AGV menggunakan pengukuran MFD

Konsep Perhitungan dari Koordinat AGV pada area magnet

Informasi: $(x_{th}, y_{th}, \theta_{th})$, (X_{pm}, Y_{pm}) $S_1 = d_1, S_2 = d_2$, dan $S_3 = d_3$

If $S_1 \sim 0$ or $S_2 \sim 0$ or $S_3 \sim 0$

If $(x_{th}, y_{th}, \theta_{th})$ is TH1

If $S_1 == A_6$ and $S_2 == A_5$ and $S_2 == A_6$

$$x_p = \frac{d_1 \sin \theta_{th}}{\sin d_1}, y_p = \frac{d_1 \sin(1.5708 - \theta_{th})}{\sin d_1}, \theta_p = \sin^{-1} \frac{x_p \sin d_1}{d_1}$$

$$x_{mp} = x_{pm} - x_p, y_{mp} = y_{pm} - y_p, \theta_{mp} = \theta_p$$

$$x_{th} = x_{mp}, y_{th} = y_{mp}, \theta_{th} = \theta_p$$

$x_{th}, y_{th}, \theta_{th}$ is TH1

Elseif $S_1 == A_6$ and $S_2 == A_6$ and $S_2 == A_5$

$$x_p = \frac{d_1 \sin \theta_{th}}{\sin d_1}, y_p = \frac{d_1 \sin(1.5708 - \theta_{th})}{\sin d_1}, \theta_p = \sin^{-1} \frac{x_p \sin d_1}{d_1}$$

$$x_{mp} = x_{pm} - x_p, y_{mp} = y_{pm} - y_p, \theta_{pm} = \theta_p$$

$$x_{mp} = x_{pm} - x_p, y_{mp} = y_{pm} - y_p, \theta_{mp} = \theta_p$$

$x_{th}, y_{th}, \theta_{th}$ is TH1

Elseif $S_1 == A_4$ and $S_2 == A_1$ and $S_2 == A_2$

$$x_p = \frac{d_1 \sin \theta_{th}}{\sin d_1}, y_p = \frac{d_1 \sin(1.5708 - \theta_{th})}{\sin d_1}, \theta_p = \sin^{-1} \frac{x_p \sin d_1}{d_1}$$

$$x_{mp} = x_{pm} - x_p, y_{mp} = y_{pm} - y_p, \theta_{pm} = \theta_p$$

$$x_{mp} = x_{pm} - x_p, y_{mp} = y_{pm} - y_p, \theta_{mp} = \theta_p$$

$x_{th}, y_{th}, \theta_{th}$ is TH0

End

End

C.5 Script LMI dari Kontroler PDC

Scripts	Ket
$A_i^T P + P A_i = -Q, \quad i = 1, 2, 3, \dots, r$	(2.43)
<pre> setlmis([]); X11=lmivar(1,[3,1]); %inisialisasi lmiterm([-1 1 1 X11],1,1); for LMM1=2:65 %A'P+PA<0 urr=LMM1-1; A11=cell2mat(AA(1,urr)); lmiterm([LMM1 1 1 X11],1,A11,'s'); end raihsaja1 = getlmis; [tmin,saja1]=feasp(raihsaja1); XX1=dec2mat(raihsaja1,saja1,X11); %ambil matriks stability P matriksp{1}=eig(XX1); %cek eigenvalue matriks stability </pre>	

Scripts LMI	Ket
Parameter yang dicari M dan $X > 0$	-
<pre> %Inisialisasi parameter LMI setlmis([]); X1=lmivar(1,[3,1]);%X1=invers(P) for LV1=1:64 %M=K*X M(1,LV1)=lmivar(1,[3,1]); end %syarat LMI pertama X>0 lmiterm([-1 1 1 X1],1,1); %X>0 </pre>	
$-X A_i^T + M_i^T B_i^T - A_i X + B_i M - 2\alpha X > 0$	(3.48)
<pre> %syarat LMI 2-65 for LM3=2:65 ur=LM3-1; AA1=cell2mat(AA(1,ur)); %ambil matriks Ai BB1=cell2mat(BB(1,ur)); %ambil matriks Bi lmiterm([-LM3 1 1 X1],-AA1,1,'s'); lmiterm([-LM3 1 1 M(1,ur)],BB1,1,'s'); lmiterm([-LM3 1 1 X1],-2*alpha,1); end </pre>	
$\begin{bmatrix} 1 & \bar{e}(0)^T \\ \bar{e}(0) & X \end{bmatrix} \geq 0$	(3.42)
<pre> %syarat LMI 66 lmiterm([-66 1 1 0],1); lmiterm([-66 2 1 0],e); lmiterm([-66 2 2 X1],1,1); </pre>	
$\begin{bmatrix} X & M_i^T \\ M_i & \mu^2 I \end{bmatrix} \geq 0$	(3.43)
<pre> %syarat LMI ke 67 - 130 for LM5=67:130 ur1=LM5-66; lmiterm([-LM5 1 1 X1],1,1); lmiterm([-LM5 2 1 M(1,ur1)],1,1); lmiterm([-LM5 2 2 0],rho.^2); end </pre>	

$\begin{bmatrix} X & XC_i^T \\ C_i X & \lambda^2 I \end{bmatrix} \geq 0$	
<pre>%syarat LMI ke 131 - 194 for LM6=131:194 lmiterm([-LM6 1 1 X1],1,1); lmiterm([-LM6 2 1 X1],C,1); lmiterm([-LM6 2 2 0],lamdha.^2); end</pre>	(3.45)
<pre>%cari nilai X>0 dan Mi raihsaja = getlmis; [tmin,saja]=feasp(raihsaja); X=dec2mat(raihsaja,saja,X1);</pre>	Hitung X
<pre>%Simpan nilai matriks Mi for flag1=1:64 MM(:, :)=dec2mat(raihsaja,saja,M(1,flag1)); Ambil{flag1}=MM; end</pre>	Matrik M
<pre>%ambil nilai gain Ki for flag2=1:64 K{flag2}=cell2mat(Ambil(1,flag2))/X; end</pre>	Gain K

Scripts LMI	Ket
$X > 0$	
<pre>setlmis([]) X2=lmivar(1,[3,1]); %inisialisasi %LMI pertama lmiterm([-1 1 1 X2],1,1); %X>0</pre>	-
$XA_i^T - M_j^T B_i^T + A_i X - B_i M_j + X A_j^T$ $- M_i^T B_j^T + A_j X - B_j M_i - 4\alpha X \leq 0, \quad i < j$	
<pre>%LMI 2-64, [i,j]-> [1,2], [2,3], ..., [63,64] for flag4=2:64 flag5=flag4-1; Ai=cell2mat(AA(1,flag5)); %matriks Ai Bi=cell2mat(BB(1,flag5)); %matriks Bi Mi=cell2mat(Ambil(1,flag5)); %matriks Mi Aj=cell2mat(AA(1,flag4)); %Aj-> i<j Bj=cell2mat(BB(1,flag4)); %Bj-> i<j Mj=cell2mat(Ambil(1,flag4)); %MjBi=Mj'*Bi';%BiMj=Bi*Mj;%MiBj=Mi'*Bj';%BjMi=Bj*Mi; MB=Mj'*Bi'+Bi*Mj+Mi'*Bj'+Bj*Mi; lmiterm([-flag4 1 1 X2],-Ai,1,'s'); %-AiX-XAi' lmiterm([-flag4 1 1 X2],-Aj,1,'s'); %-AjX-XAj' lmiterm([-flag4 0 0 0],MB); % lmiterm([-flag4 1 1 X2],-4*alpha,1); %-4*alpha*X end</pre>	(3.49)
<pre>raihsajal = getlmis; [tmin,sajal]=feasp(raihsajal); %ambil matriks stability P PP=dec2mat(raihsajal,sajal,X11); %cek eigenvalue matriks stability P PPl=eig(PP);</pre>	Ambil nilai P

C.6 Script Perhitungan Matriks parameter Q dan Pengujian Nilai

Scripts LMI	Ket
$A_i^T P + P A_i = -Q, i = 1, 2, 3, \dots, r$ $V = -\bar{e}^T P \bar{e} > 0, \quad t \geq 0$ $\dot{V} = -\bar{e}^T Q e < 0, \quad t \geq 0$	(3.63)
<pre> QQQ=[11.6539 -0.1321 -0.1404 -0.1321 15.0762 -2.4120 -0.1404 -2.4120 11.5525]; urut=1; for ex1=0.1:0.1:4 for ey1=0.1:0.1:4 for eth1=0.1:0.1:4 ebar=[ex1;ey1;eth1]; %ebar=[ex1;ey1;-eth1]; %ebar=[ex1;-ey1;-eth1]; %ebar=[-ex1;-ey1;-eth1]; Vdot=-ebar'*QQQ*ebar; simpanVdot(1,urut)=Vdot; urut=urut+1; end end end end </pre>	
<pre> PPP= [1.1496 -0.1727 -0.1005 -0.1727 63.0547 30.1846 -0.1005 30.1846 15.7346]; urut1=1; for ex1=0.1:0.1:4 for ey1=0.1:0.1:4 for eth1=0.1:0.1:4 ebar=[ex1;ey1;eth1]; %ebar=[ex1;ey1;-eth1]; %ebar=[ex1;-ey1;-eth1]; %ebar=[-ex1;-ey1;-eth1]; Vs=ebar'*PPP*ebar; simpanV(1,urut1)=Vs; urut1=urut1+1; end end end end </pre>	
$V = [19.425 \quad 6.46 \quad 24.965 \quad 74.93 \quad 156.38 \quad 269.295 \quad 413.67]$	
$\dot{V} = [-28652.47 \quad -29814.17 \quad -30998.976 \quad -32206.884]$	

C.7 Script Pengujian 4.2.1: Pengujian Kontroler Pada Model Kinematika

%preallocation memory
<pre> Ambil=cell(1,64); M=zeros(1,64); AA=cell(1,64); BB=cell(1,64); K=cell(1,64); matriksp=cell(1,3); </pre>
%inisialisasi

<pre> D=0; C=[1 0 0;0 1 0;0 0 1]; I=[1 0 0;0 1 0;0 0 1]; %posisi&sudut awal referensi x0r=5;y0r=5;th0r=45; %posisi awal robot x0=4.5;y0=5.5;th0=0; %inisialisasi error awal ex0=x0r-x0; </pre>	<pre> ey0=y0r-y0; eth0=deg2rad(th0r-th0); theta=deg2rad(th0); RM=[cos(theta) -sin(theta) 0 sin(theta) cos(theta) 0 0 0 1]; %matrik rotasi e_XY=[ex0;ey0;eth0]; e=RM*e_XY; </pre>
%Pengaturan variabel premise	
<pre> ab=1; loop1=1; %wr loop2=1.06; %vr loop3=0.4; %ex loop4=0.2; %ey loop5=1; %sineth loop6=0.1; %vr_ey^2 loop7=2; %rho loop8=1.1; %lambda %parameter LMI alpha=0.9; rho=loop7*I; lamdha=loop8*I; </pre>	<pre> %send to simulink z1=loop1;z11=-loop1; z2=loop2;z22=-loop2; z3=loop3;z33=-loop3; z4=loop4;z44=-loop4; z5=loop5;z55=-loop5; z6=loop6;z66=0; %nilai max min amax=loop1;amin=-loop1; bmax=loop2;bmin=1.06; cmax=loop3;cmin=-loop3; dmax=loop4;dmin=-loop4; emax=1;emin=2/pi; fmax=loop6;fmin=0; </pre>
%Matriks Linearisasi Ai dan Bi	
<pre> aaa=1; for ii=1:2 if mod(ii,2)==1;zz1=amin; else;zz1=amax; end for jj=1:2 if mod(jj,2)==1;zz2=bmin; else;zz2=bmax; end for kk=1:2 if mod(kk,2)==1;zz3=cmin; else;zz3=cmax; end for ll=1:2 if mod(ll,2)==1;zz4=dmin; else;zz4=dmax; end for mm=1:2 if mod(mm,2)==1;zz5=emin; else;zz5=emax; end for nn=1:2 if mod(nn,2)==1;zz6=fmin; else;zz6=fmax; end </pre>	<pre> A(:,:)= [0 zz1 0 -zz1 0 zz2*zz5 0 0 0]; B(:,:)= [-zz3 zz6 zz2*zz4*zz5 0 -zz2*zz3*zz4 -zz2*zz3*zz5 0 -zz2*zz4 -zz2*zz5]; AA{aaa}=A; BB{aaa}=B; aaa=aaa+1; end end end end end end </pre>
%LMI	
<pre> setlmis([]); Xl1=lmivar(1,[3,1]); %inisialisasi matriks stability P lmitem([-1 1 1 Xl1],1,1); for LMM1=2:65 %A'P+PA<0 urr=LMM1-1; A11=cell2mat(AA(1,urr)); lmitem([LMM1 1 1 Xl1],1,A11,'s'); end raihsaja1 = getlmis; </pre>	

```

[tmin,saja1]=feasp(raihaja1);
XX1=dec2mat(raihaja1,saja1,X11); %ambil matriks stability P
matriksp{1}=eig(XX1); %cek eigenvalue matriks stability P
%LMI start
setlmis([]);
%inisialisasi
X1=lmivar(1,[3,1]);%X1=invers(P)
for LV1=1:64 %M=K*X
    M(1,LV1)=lmivar(1,[3,1]);
end
%LMI pertama
lmiterm([-1 1 1 X1],1,1); %X>0
for LM3=2:65
    ur=LM3-1;
    AA1=cell2mat(AA(1,ur));
    BB1=cell2mat(BB(1,ur));
    lmiterm([-LM3 1 1 X1],-AA1,1,'s');
    lmiterm([-LM3 1 1 M(1,ur)],BB1,1,'s');
    lmiterm([-LM3 1 1 X1],-2*alpha,1);
end
lmiterm([-66 1 1 0],1);
lmiterm([-66 2 1 0],e);
lmiterm([-66 2 2 X1],1,1);
for LM5=67:130
    ur1=LM5-66;
    lmiterm([-LM5 1 1 X1],1,1);
    lmiterm([-LM5 2 1 M(1,ur1)],1,1);
    lmiterm([-LM5 2 2 0],rho.^2);
end
for LM6=131:194
    lmiterm([-LM6 1 1 X1],1,1);
    lmiterm([-LM6 2 1 X1],C,1);
    lmiterm([-LM6 2 2 0],lamdha.^2);
end
%ambil nilai LMI
raihaja = getlmis;
[tmin,saja]=feasp(raihaja);
X=dec2mat(raihaja,saja,X1);
XX2=inv(X);
matriksp{2}=eig(XX2);
%simpan nilai M
for flag1=1:64
    MM(:,:)=dec2mat(raihaja,saja,M(1,flag1));
    Ambil{flag1}=MM;
end

% Persamaan 3.49
setlmis([])
X2=lmivar(1,[3,1]);
%LMI pertama
lmiterm([-1 1 1 X2],1,1); %X>0
%LMI 2-64, [i,j]-> [1,2], [1,3], ..., [1,64]
for flag4=2:64
    flag5=flag4-1;
    Ai=cell2mat(AA(1,flag5)); %matriks Ai
    Bi=cell2mat(BB(1,flag5)); %matriks Bi
    Mi=cell2mat(Ambil(1,flag5)); %matriks Mi

```

<pre> Aj=cell2mat(AA(1,flag4)); %Aj-> i<j Bj=cell2mat(BB(1,flag4)); %Bj-> i<j Mj=cell2mat(Ambil(1,flag4)); %MjBi=Mj'*Bi';%BiMj=Bi*Mj;%MiBj=Mi'*Bj';%BjMi=Bj*Mi; MB=Mj'*Bi'+Bi*Mj+Mi'*Bj'+Bj*Mi; lmiterm([-flag4 1 1 X2],-Ai,1,'s'); %-AiX-XAi' lmiterm([-flag4 1 1 X2],-Aj,1,'s'); %-AjX-XAj' lmiterm([-flag4 0 0 0],MB); % lmiterm([-flag4 1 1 X2],-4*alpha,1); %-4*alpha*X end raihsaja2 = getlmis; [tmin,saja2]=feasp(raihsaja2); X3=dec2mat(raihsaja2,saja2,X2); XX3=inv(X3); matriksp{3}=eig(XX3); %Simpan nilai K for flag2=1:64 K{flag2}=cell2mat(Ambil(1,flag2))/X; End </pre>	
%jalankan simulink dan ambil data	
<pre> sim('kinematik_64_straight'); t = ScopeData1.time; data1=ScopeData1.signals.values(:,:); xr= data1(:,1); yr= data1(:,2); xc = simout.signals.values(:,:); yc = simout1.signals.values(:,:); ex= xyth_saja(:,2); ey= xyth_saja(:,3); etheta= xyth_saja(:,4); vvc= vw(:,2); wwc= vw(:,3); hold off </pre>	
%Hitung MSE	
<pre> hitu1=zeros(2001,1); hitu2=zeros(2001,1); hitu3=zeros(2001,1); for hit1=1:2001 mse=ex(hit1,1).^2; hitu1(hit1)=mse; end for hit2=1:2001 mse2=ey(hit2,1).^2; hitu2(hit2)=mse2; end </pre>	<pre> for hit3=1:2001 mse3=etheta(hit3,1).^2; hitu3(hit3)=mse3; end MSE_ex=sum(hitu1(:,1))/2001; MSE_ey=sum(hitu2(:,1))/2001; MSE_etheta=sum(hitu3(:,1))/2001; </pre>
%Cek Respon	
<pre> stepinfo(ex,t) stepinfo(ey,t) stepinfo(etheta,t) </pre>	
%Plot respon	
<pre> figure(ab);hold on plot(xr,yr,'--*','MarkerIndices',1:length(t)-1:length(t)) axis([0 25 0 25]); % axis([4.2 6 4.4 6]); %title('Hasil tracking circle trajectory') plot(xc,yc,'-o','MarkerIndices',1:length(t)-1:length(t)) </pre>	

```

% xe=[0.3 0.45];
% ye=[0.3 0.4];
% cx=[0.2 0.25];
% cy=[0.4 0.65];
% satu=annotation('textarrow',xe,ye,'String','reference start')
% dua=annotation('textarrow',cx,cy,'String','vehicle start')
ylabel('y(m)')
xlabel('x(m)')
legend('reference','vehicle')
figure(2)
plot(t,ex,'--',t,ey,':',t,etheta)
ylabel('e_x(m),e_y(m),e_t_h(rad)')
xlabel('t(s)')
legend('error x','error y','Error theta')
figure(3)
plot(t,vvc,'--',t,wwc,':')
ylabel('v_c(m/s),w_c(m/s)')
xlabel('t(s)')
legend('kecepatan linier','kecepatan angular')

```

C.8 Lampiran Script untuk Pengujian 4.3.2 berdasarkan [11]

%inisialisasi	
D=0; C=[1 0 0;0 1 0;0 0 1]; I=[1 0 0;0 1 0;0 0 1]; %posisi&sudut awal referensi x0r=5;y0r=0;th0r=0; %posisi awal robot x0=5;y0=1;th0=0; %inisialisasi error awal ex0=x0r-x0;	ey0=y0r-y0; eth0=deg2rad(th0r-th0); theta=deg2rad(th0); RM=[cos(theta) -sin(theta) 0 sin(theta) cos(theta) 0 0 0 1]; %matrik rotasi e_XY=[ex0;ey0;eth0]; e=RM*e_XY;
%Pengaturan variabel premise	
loop(1)=0.1; %wr loop(2)=1.4142; %vr loop(3)=0.1; %ey loop(4)=0.1; %ex loop(5)=0.4; %sineth loop(6)=0.1; %rho alpha=0.6; %send to simulink z1=loop1;z11=-loop1; z2=loop2;z22=-loop2;	z3=loop3;z33=-loop3; z4=loop4;z44=-loop4; z5=loop5;z55=-loop5; %nilai max min nmax(1)=loop(1);nmin(1)=-loop(1); nmax(2)=loop(2);nmin(2)=loop22; nmax(3)=loop(3);nmin(3)=-loop(3); nmax(4)=loop(4);nmin(4)=-loop(4); nmax(5)=loop(5);nmin(5)=loop55;
%Matriks Linearisasi Ai dan Bi	
%Untuk menyingkat for i=1-5 sama seperti pada Lampiran C.7 A(:,:)= [0 zz(1) 0;-zz(1) 0 zz(3);0 0 0]; B(:,:)= [-1 zz(3);0 -zz(4);0 -zz(5)]; AA{aaa}=A; BB{aaa}=B; aaa=aaa+1;	
%LMI	
setlmis([]); %inisialisasi X1=lmivar(1,[3,1]);%X1=invers(P) for LV1=1:32 %M=K*X M(1,LV1)=lmivar(2,[2,3]);	

```

end
%LMI pertama
lmiterm([-1 1 1 X1],1,1);      %X>0
%LMI ketiga
for LM3=2:33
    ur=LM3-1;
    AA1=cell2mat(AA(1,ur));
    BB1=cell2mat(BB(1,ur));
    lmiterm([-LM3 1 1 X1],-AA1,1,'s');
    lmiterm([-LM3 1 1 M(1,ur)],BB1,1,'s');
    lmiterm([-LM3 1 1 X1],-2*alpha,1);
end
%LMI keempat
lmiterm([-34 1 1 0],1);
lmiterm([-34 2 1 0],e);
lmiterm([-34 2 2 X1],1,1);

%LMI kelima
for LM5=35:66
    url=LM5-34;
    lmiterm([-LM5 1 1 X1],1,1);
    lmiterm([-LM5 2 1 M(1,url)],1,1);
    lmiterm([-LM5 2 2 0],rho.^2);
end
%ambil nilai LMI
raihsaja = getlmis;
[tmin,saja]=feasp(raihsaja);
X=dec2mat(raihsaja,saja,X1);
% XX=inv(X);
for flag1=1:32
    MM(:,:)=dec2mat(raihsaja,saja,M(1,flag1));
    Ambil{flag1}=MM;
end

for flag2=1:32
    K{flag2}=cell2mat(Ambil(1,flag2))/X;
end
end

%jalankan simulink dan ambil data
sim('paper_kine32_persegi');
t = ScopeData1.time;
data1=ScopeData1.signals.values(:,:);
xr= data1(:,1);
yr= data1(:,2);
xc = simout.signals.values(:,:);
yc = simout1.signals.values(:,:);
ex= xyth_saja(:,2);
ey= xyth_saja(:,3);
etheta= xyth_saja(:,4);
vvc= vw(:,2);
wwc= vw(:,3);
vvr= vwr(:,2);
wwr= vwr(:,3);

%Program Plot, perhitungan MSE, dan Respon sama seperti Lampiran
C.7

```

DAFTAR INDEX

algoritma navigasi, 5
controllability, 30
differential drive, 9
dinamika sistem, 31
DOB, 37
dynamic error, 43
extended state observer, 11
fungsi Lyapunov, 45
fuzzy model, 20
Fuzzy T-S Model, 14
input kontrol, 43
karakteristik polinomial, 44
kecepatan angular ODMR, 17
kecepatan linier dan angular, 27
kestabilan Routh, 45
koefisien array, 44
koreksi error posture, 29
Koreksi heading ODMR, 18
Lie Bracket, 31
LMI, 22, 42
magnetic compass, 16
magnetic dipole, 6
magnetic spot, 5
magnitude, 6
matrik linearisasi, 40
Matriks linearisasi, 15
matriks stabilitas, 43
MFD, 33
model Dinamika, 31
Model error, 10
model *error* kinematika, 12
Model error kinematika baru, 13
Model extended, 13
Nonholonomic, 27
Output akhir fuzzy model, 20
PDC, 22
pemodelan magnet, 33
persamaan differensial, 10
Posisi dan orientasi, 6
positif definit, 41
steady state error, 44
weighted average, 21

Halaman ini sengaja dikosongkan



Mamat Septyan dilahirkan pada tahun 1989 di Nganjuk, Jawa Timur, Indonesia. Penulis menempuh Pendidikan SMA pada tahun 2008, kemudian melanjutkan ke jenjang D4 dan memperoleh gelar Sarjana Sains Terapan pada tahun 2013. Setelah itu penulis terjun ke dunia kerja terutama pada industri alat berat. Pada tahun 2016 penulis melanjutkan studi program Magister di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember dengan bidang keahlian Teknik Sistem Pengaturan. Pada saat ini penulis sedang menyelesaikan program magister pada bulan Januari 2019.

Halaman ini sengaja dikosongkan